STRUCTURE IN MODERN DATA AND HOW TO EXPLOIT IT: SOME SIGNAL PROCESSING APPLICATIONS

By

MUHAMMAD ASAD LODHI

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Prof. Waheed U. Bajwa

And approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

March 2020

ABSTRACT OF THE DISSERTATION

STRUCTURE IN MODERN DATA AND HOW TO EXPLOIT IT: SOME SIGNAL PROCESSING APPLICATIONS

by Muhammad Asad Lodhi

Dissertation Director: Prof. Waheed U. Bajwa

Modern applications in real-world scenarios generate data that are massive and often times highly structured. Exploiting this structure in an effective manner leads to improved performance, and reduced computational and memory complexities. Moreover, successful exploitation of this underlying structure also admits efficient data representation, superior inference capabilities, and scalable estimation with fewer samples. This dissertation investigates these advantages of structure exploitation in three applications: (i) signal detection and classification under the union-of-subspaces model, (ii) learning product graphs underlying smooth graph signals, and (iii) distributed radar imaging under position errors and unsynchronized clocks.

For detection under the union-of-subspaces model we derive the generalized likelihood ratio tests and bounds on the recovery performance under varying levels of knowledge about colored noise in the observations. We also make explicit the dependence of the performance metrics on the geometry of the subspaces comprising the union and of the colored noise. We validate the theoretical insights through numerical experiments on synthetic and real data.

In regards to the product graph learning problem, we devise a method to learn structured graphs from data that are given in the form of product graphs. Product graphs arise nat-

urally in many real-world datasets and provide an efficient and compact representation of large-scale graphs through several smaller factor graphs. We initially pose the graph learning problem as a linear program, which (on average) outperforms the state-of-the-art graph learning algorithms. Afterwards, we devise an alternating minimization-based algorithm aimed at learning various types of product graphs from data, and establish local convergence guarantees to the true solution. Finally the superior performance and reduced sample complexity of the proposed algorithm over existing methods are also validated through numerical simulations on synthetic and real datasets.

Our final focus is on distributed radar imaging, which is essential for modern radar applications to enable high resolution imaging through a large synthetic aperture. This distributed setup suffers from two commons problems: (i) access to imprecise antenna locations, and (ii) clock mismatch between the distributed components, which adversely affects the final reconstruction of radar scene. We develop exact models to address both of these issues in the most general settings by modeling the errors as convolutions with 1-sparse spatial and temporal shifts. The radar scene reconstruction problems associated with the resulting forward models can then be expressed as nonconvex blind deconvolution problems, which can be solved through a block coordinate descent-based method. At each step of this method, each subproblem is convex and can be solved using accelerated proximal gradient methods like FISTA. Finally, we characterize the theoretical performance of the proposed method by deriving error bounds for the estimated unknowns, and through numerical simulations on synthetic data obtained under varying degrees of noise in the observations.

ACKNOWLEDGEMENTS

I would like to start, first and foremost, by thanking my advisor Prof. Waheed Bajwa who was critical of my abilities and my work, but also gave me free reign to explore my interests and choose my own directions. Under his supervision I have grown academically and professionally, and I owe any and all achievements under my belt to his guidance and support. I extend to him my absolute appreciation and thanks. I would also like to thank my committee members Prof. Athina Petropulu, Prof. Kristin Dana, and Prof. Yuqian Zhang for their insights towards my work and for making time for me in the first place. I also thank Dr. Petros Boufounos from MERL first for serving as an outside committee member. He deserves my gratitude in more than one ways for being an awesome internship advisor and for his utmost support over the last couple of years. I must also extend my thanks towards his colleagues Dr. Hassan Mansour and Dr. Yanting Ma for their constant help and guidance during my internships. I would also like to thank the administrative staff at the department of Electrical and Computer Engineering at Rutgers for their tireless help over my time at Rutgers.

Moving on to my friends and labmates, two people have been their with me from the start (even if their journey ended before mine) and have helped me in more ways than I can think of: Haroon Raja and Talal Ahmed. Thank you for tolerating me and my strong-headedness in my earlier years as a graduate student. For someone who started their graduate journey so recently, I have never seen anyone so driven, persevering and intelligent as Batoul Taki. I thank you for your friendship and your support in my final years of graduate life. And I must also thank Arpita Gang, who along with Batoul, has been an amazing friend and has made my time at Rutgers much more interesting and bearable.

Finally, words would not suffice for what I want to say to my two pillars of love and support. My parents. I cannot express how grateful I am to you, and how grateful I am to Allah for making me your son.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Motivation

Modern applications in real-world scenarios generate data that is often times highly structured. Moreover, in numerous applications there is an inherent structure in the data acquisition and the data generation processes. Exploiting the structure in aspects of data generation and acquisition, and in the data itself, has been at the core of countless information processing and learning methodologies. A (possible) categorization of these methods is shown in Fig. 1.1.

Although structure exploiting approaches have been investigated in the past, the interest in this field got renewed with the advent of compressed sensing (CS) [1, 2, 3], which aims to reconstruct a sparse signal from fewer than Nyquist-dictated samples. CS also sparked researchers in the field to explore and exploit structure in the multidimensional regime (consisting of matrices and tensors), with the aim of generalizing methods initially designed for vector-valued data. Independent of this direction of research, there has been increasing interest in graph signal processing that aims to generalize the existing approaches in classical signal processing to data that lives on structured domains [4, 5, 6, 7, 8]. Graph signals provide a natural way to represent data and subsume the classical way of representing signals. Exploiting structure in an effective manner has been shown to result in parsimonious data representation, superior inference capabilities, and reduced computational and memory complexities in numerous applications. Moreover, given the massive nature of modern data, it is imperative to develop scalable and efficient structure exploiting techniques to further entertain the advantages entailed.

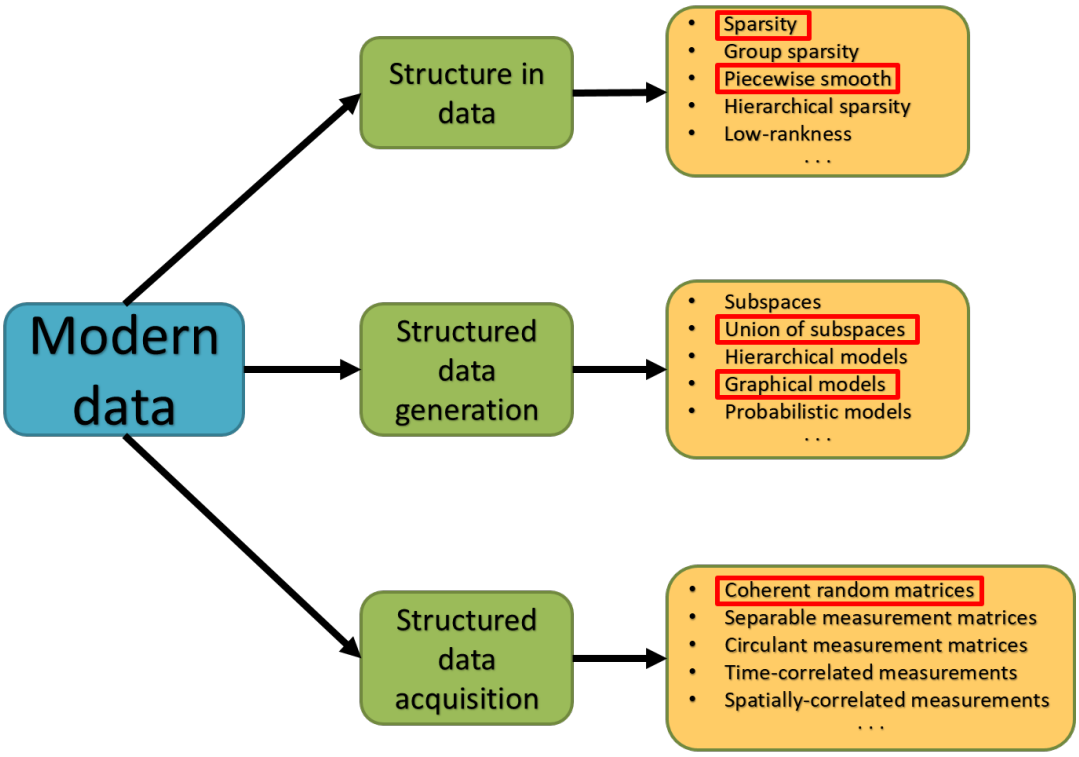

Figure 1.1: A (possible) high-level representation of structure exploiting methodologies in modern data science. All three blocks are interrelated. The contributions of this dissertation are indicated by red boxes in the figure.

1.2 Research contributions

Our main contributions towards structure exploitation in data, data acquisition, and data generation processes in this dissertation are presented below.

1.2.1 Detection theory for union of subspaces

Signal detection is one of the oldest problems in signal processing with a rich literature under the subspace model [9, 10]. However, recently, the focus has shifted to a *non-linear* model named union of subspaces (UoS) model [11, 12, 13, 14], which dictates that real-world data is generated by a collection of subspaces rather than just one subspace. Our work in Chapter. 2 focused on formulating a theory for signal detection under the UoS model. We formalized the problems of signal detection and active subspace classification problems

under the UoS model, when the data is generated by only one subspace in one instance. We posed the detection and classification problems as binary and multiple hypothesis tests and presented the generalized likelihood ratio tests for each. We characterized the performance of the proposed tests in terms of bounds on the probabilities of false alarm, detection, and correct classification (active subspace detection). Furthermore, we analyzed these bounds in light of the geometry between the subspace and the geometry of the colored noise in the observed signal. We showed that the probability of correctly identifying the active subspace increases with the increasing principal angles between the subspaces comprising the union. We also demonstrated, for same noise levels, the subspaces that live close to the higher-order eigenvectors of the colored noise covariance (i.e., eigenvectors corresponding to higher eigenvalues) have lower detection probabilities, and vice versa. We validated the performance and the analytical insights through numerical experiments on synthetic and real datasets.

1.2.2 Learning structured graphs from data

Real-world data is often times associated with irregular structures that can analytically be represented as graphs. Having access to this graph, which is sometimes trivially evident from domain knowledge, provides a better representation of the data and facilitates various information processing tasks. However, in cases where the underlying graph is unavailable, it needs to be learned from the data itself for data representation, data processing and inference purposes [7, 15, 16]. Existing literature on learning graphs from data has mostly considered arbitrary graphs [15, 16], whereas the graphs generating real-world data tend to have additional structure that can be incorporated in the graph learning procedure. Structure-aware graph learning methods require learning fewer parameters and have the potential to reduce computational, memory and sample complexities. In light of this, our work in Chapter. 3 devised a method to learn structured graphs from data that are given in the form of product graphs [6]. Product graphs arise naturally in many real-world datasets

and provide an efficient and compact representation of large-scale graphs through several smaller factor graphs. To this end, first the graph learning problem was posed as a linear program, which (on average) outperformed the state-of-the-art graph learning algorithms. This formulation is of independent interest itself as it shows that graph learning is possible through a simple linear program. Afterwards, an alternating minimization-based algorithm aimed at learning various types of product graphs was proposed, and local convergence guarantees to the true solution were established for this algorithm. Finally the performance gains, reduced sample complexity, and inference capabilities of the proposed algorithm over existing methods were also validated through numerical simulations on synthetic and real datasets.

1.2.3 Distributed radar imaging under ambiguous array parameters

Distributed radar imaging is essential for modern radar applications where high resolution imaging is enabled through a large synthetic aperture by combining several small aperture antennas. However, two commons problems faced in distributed radar are (i) imprecise knowledge of antenna locations, and (ii) clock mismatch between the distributed components, which adversely effects the final reconstruction performance of radar scene. In Chapter. 3 of this dissertation, we developd exact models to address both of these issues in the most general settings, i.e., instead of modeling the position and clock errors as a combined complex phase and gain vector (as done traditionally), we model them separately in the image and time domains as convolutions with 1-sparse spatial and temporal shifts. The radar scene reconstruction problems associated with the resulting forward models could then be expressed as blind deconvolution problems in two or more unknowns. We proposed a block coordinate descent-based algorithm to solve these nonconvex blind deconvolution problems, where each subproblem was convex and was solved using accelerated proximal gradient methods like FISTA. For theoretical characterization of the performance, we derived bounds on the error of estimated unknowns from their true values for all problems

posed in this work. We also validate the performance of our proposed forward models and the reconstruction algorithm through numerical simulations with synthetic data under varying degrees of noise in the observations.

1.2.4 Notation

The following notational convention is used throughout this dissertation. We use bold lower-case and bold-upper case letters to represent vectors and matrices, respectively. Calligraphic letters are used to represent tensors, which are arrays of three or more dimensions. Given a vector $\mathbf{a}$, $\|\mathbf{a}\|_p$ denotes its ℓ_p-norm and $|\mathbf{a}|$ denotes its elementwise absolute values. For a matrix $\mathbf{A}$, $\mathbf{a}_j$ and a_{ij} denote its j-th column and (i,j)-th entry, respectively. Further, $\|\mathbf{A}\|_F$ represents its Frobenius norm, $\|\mathbf{A}\|$ represents its spectral norm, $\mathbf{A}^\dagger$ represents its Moore-Penrose inverse, and finally $|\mathbf{A}|$ denotes its determinant. Moreover, $\|\mathbf{A}\|_1$ represents the ℓ_1-norm of the entries of $\mathbf{A}$, while $\|\mathbf{A}\|_{1,\text{off}}$ represents the ℓ_1-norm of the off-diagonal entries of $\mathbf{A}$. The Kronecker and Cartesian products of two matrices $\mathbf{A}$ and $\mathbf{B}$ are denoted by $\mathbf{A}\otimes\mathbf{B}$ and $\mathbf{A}\oplus\mathbf{B}$, respectively [17]. The strong product of two matrices, which is the sum of Cartesian and Kronecker products, is denoted by $\mathbf{A}\boxtimes\mathbf{B}$. Furthermore, $\underset{\mathbf{s}}{\otimes}$, $\underset{\mathbf{s}}{\oplus}$, and $\underset{\mathbf{s}}{\boxtimes}$, respectively denote the Kronecker, Cartesian, and strong products taken over the indices provided by the entries of the vector $\mathbf{s}$. The Hadamard product (elemntwise product) of two vectors or matrices is denoted by "$\circ$". For a tensor $\mathcal{T}$, $\mathcal{T}_{(i)}$ represents its matricization (flattening) in the i-th mode and $\mathrm{vec}(\mathcal{T})$ represents its vectorization [18]. Also, ":" represents the scalar product or double dot product between two tensors, which results in a scalar [18]. Finally, $\times_i$ represents matrix multiplication in the i-th mode of a tensor and $\underset{\mathbf{s}}{\times}$ represents matrix multiplications in the modes of a tensor specified in the entries of the vector $\mathbf{s}$. In chapter 4, calligraphic letters are reserved for representing operators that act on (one or more) vectors and matrices.

We use $\mathrm{diag}(\mathbf{x})$ to denote a diagonal matrix with diagonal entries given by the entries of the vector $\mathbf{x}$, $\mathbf{1}$ to denote a vector of all ones with appropriate length, and $\mathbb{1}$ to denote

a tensor of all ones of appropriate size. We denote the set with elements $\{1, 2, \ldots, K\}$ as $[K]$, and $[K] \setminus k$ represents the same set without the element k. The sets of valid Laplacian and weighted adjacency matrices are represented by $\mathcal{L}$ and $\mathcal{W}$, respectively. The set of weighted adjacency matrices with any product structure is denoted by $\mathcal{W}_p$. We use the standard "big-$\mathcal{O}$" notation to denote asymptotic scaling. Finally, $Q(\cdot)$, $\Gamma(\cdot)$, and $K_n(\cdot)$ denote the Gaussian Q function, the Gamma function, and the modified Bessel function of the second kind with parameter n, respectively.

1.2.5 Organization

In the following chapters we will detail our contributions under the three problems described in this section. We will formulate each problem followed by our proposed methods for solving each problem, accompanied thereafter by theoretical performance guarantees of the proposed methods. We will also validate the performance of each proposed method through numerical simulations on synthetic and real data.

CHAPTER 2
DETECTION THEORY FOR UNION OF SUBSPACES

2.1 Introduction

Detection theory has a long history in the signal processing literature. Classical detection theory is often based on the *subspace model*, in which the signal to be detected is assumed to come from a low-dimensional subspace embedded in a high-dimensional ambient space [9, 10]. However, recently a nonlinear generalization of the subspace model, termed the *union of subspaces* (UoS) model [11, 12, 13, 14], has gained attention in the literature due to its ability to better model real-world signals. Indeed, data in many real-world scenarios tend to be generated by processes that switch/operate in different modes. In such instances, data generated through each mode of the process can be modeled as lying on a subspace, in which case the entire data generated during the process as a whole can be best described as coming from a union of subspaces [19, 20, 21, 22, 23, 24, 25]. Some specific instances of such processes include: (i) radar target detection involving multiple targets, with only one target being present at a time and each target being characterized by its own specific spectral signature; (ii) user detection in a wireless network, with only one user transmitting at a time and each user having its own transmit codebook; and (iii) image-based verification of employees in an organization, with the verification system using a database of employees' facial images collected under varying lighting conditions.

Broadly speaking, and under the assumption of processes following the UoS model, we focus on the following questions in this work: (i) whether an observed signal (e.g., spectral data, radio frequency (RF) observations, or an image) corresponds to a known generation mechanism (e.g., spectral signatures of known targets, RF transmissions from known users, or faces of known employees); and (ii) which mode (e.g., which known target,

which known user, or which existing employee) from the known generation mechanism gave rise to the observed signal. In this context, we revisit in this chapter the problem of detection of signals under various additive noise models for the case when the signal conforms to the UoS model. Our goals in this regard are: (i) derivation of tests for detection of both the signal and the underlying active subspace (mode), and (ii) characterization of the performance of these tests in terms of geometry of the subspaces.

2.1.1 Prior work

There exists a rich body of literature concerning detection of signals under the subspace model; see, e.g., [27, 28, 29, 30]. The most well-studied method in this regard is the matched subspace detector [27], which projects the received signal onto the subspace of interest and compares its energy against a threshold. A naïve approach to detection under the UoS model would be to treat it as a subspace detection problem by replacing the union with direct sum and using the resulting subspace within the matched subspace detector. However, such an approach not only results in high false alarm rates (for obvious reasons), but it also does not enable detection of the active subspace. A better alternative is to treat the detection problem as a multiple hypothesis testing problem, as in [23], with each test given by an individual matched subspace detector. We establish in this chapter that such an approach will have the same performance as a generalized likelihood ratio test (GLRT) for the case of a single active subspace.

Recently, there have been a few works that are directly related to the detection problem under the UoS model [1, 2, 3, 24, 26]. One of the biggest differences between these (and related) works and this chapter is that the existing works cannot explain the variability of detection performance under the UoS model for different problems with same problem parameters (e.g., number and dimension of subspaces, and signal-to-noise ratio); see, e.g., Fig. 2.15 and the accompanying discussion. In contrast, we have been able to establish in this chapter that such variability is a quantifiable function of the geometry (expressed

Work	***Framework***	***Gaussian Noise Model***	***Signal Detection***	***Active Subspace Detection***	***Impact of Geometry***
[1], [2],[3]	compressive sensing	white, w/ known variance	✓	✗	✗
[24]	general UoS	colored, w/ known cov. and unknown var.	✓	✓	✗
[26]	linear sampling of UoS	white, w/ known var.	✗	✓	✗
This work	general UoS	colored, w/ known statistics colored, w/ partially unknown statistics colored, w/ completely unknown statistics	✓	✓	✓

Table 2.1: A brief comparison of this work with related prior works in the literature

in terms of principal angles) of individual subspaces in the union and the geometry of the noise.

In terms of explicit comparisons with individual works related to this chapter, [1] studies the problem of signal detection under the compressive sensing framework [31], with the final results involving analysis of a GLRT for a binary hypothesis test. These *compressive detection* results can be considered a special instance of those for detection under the UoS model, since a sparse signal can be thought of as lying in a union of (exponentially many) subspaces [11]. The nature of these results, however, does not enable understanding of the general detection problem under the UoS model, especially in relation to geometry of the underlying subspaces. First, individual subspaces do not explicitly appear in compressive detection; rather, the results are presented in terms of the so-called "measurement matrix," which obfuscates the role of individual subspaces in detection performance. Second, the most useful of compressive detection results involve the use of *random* measurement matrices; translated into the UoS model, this corresponds to randomly generated subspaces. Since random subspaces tend to be equiangular (with high probability), compressive detection literature does not lend itself to understanding the role of subspace geometry in signal detection. Similar to [1], [2] also studies the compressive detection problem, but in the context of radar-based multi-target detection. While the analysis in [2] is based on the use of the LASSO [32] for detection, it too does not offer geometric insights into the general UoS-based detection problem. In [3], the authors extend the original compressive detection framework of [1] to more general settings, but the final results are still couched in terms of the sparsity framework and they fail to bring out the geometric interplay between the different subspaces.

The work that is most closely related to ours is [24], in which the authors study the signal and the active subspace detection problems under the UoS framework in the context of radar target detection. The (signal and active subspace) detection schemes proposed in [24] are based on multiple hypothesis testing. The analysis in [24] is for the case of

colored Gaussian noise with unknown variance but *known* covariance matrix. Further, since the analysis is in terms of the spectral signatures of targets, it does not help understand the interplay between the detection performance and the geometry of subspaces. Finally, [24] does not investigate invariance properties of the derived test statistics.

Recently, [26] has studied both recovery of a signal conforming to the UoS model and detection of the corresponding active subspace in the presence of a linear sampling operator. This work, however, is fundamentally focused on understanding the role of the sampling operator within the active subspace detection problem. Further, it assumes white Gaussian noise with known variance, does not investigate the related problem of signal detection, and does not focus on the geometry of subspaces as an integral component of the detection problem.

2.1.2 Our contributions

The major contributions of this chapter include derivation, analysis, and understanding of various GLRTs for the signal and the active subspace detection problems under the UoS model for different noise settings. One of our main contributions in this regard is a comprehensive understanding of the two detection problems in terms of characterization of the performance of the derived GLRTs through the probabilities of detection, classification, and false alarm, geometry of the underlying subspaces, and invariance properties of the test statistics. One of the key insights of this work is that the probability of correct identification of the active subspace increases with increasing principal angles between subspaces in the union. While this makes intuitive sense, our analysis provides theoretical justification for such an assertion. Further, our work also helps understand the relationship between a binary and a multiple hypothesis testing approach to the signal detection problem under the UoS model. Finally, we provide extensive numerical experiments to highlight the usefulness of our analysis and its superiority to prior works such as [26]. We refer the reader to Table 2.1 for a brief comparison of our work with existing literature.

2.1.3 Organization

The rest of the chapter is organized as follows. In Sec. 4.2, we formulate the signal and the active subspace detection problems under the UoS model. Sec. 2.3 derives and analyzes the GLRTs for these two problems under different noise conditions. Sec. 2.4 provides a discussion of the results obtained in Sec. 2.3. Sec. 2.5 presents the results of numerical experiments on both synthetic and real-world data, while we conclude the chapter in Sec. 2.6.

2.2 Problem formulation

We study two interrelated detection problems in this chapter. The first one, referred to as *signal detection*, involves deciding between an observation $\mathbf{y} \in \mathbb{R}^m$ being just noise or it being an unknown signal $\mathbf{x} \in \mathbb{R}^m$ embedded in noise. Mathematically, this can be posed as a binary hypothesis test with the null ($\mathcal{H}_0$) and the alternate ($\mathcal{H}_1$) hypotheses given by:

$$\begin{aligned} \mathcal{H}_0 : &\quad \mathbf{y} = \mathbf{n}; \\ \mathcal{H}_1 : &\quad \mathbf{y} = \mathbf{x} + \mathbf{n}; \end{aligned} \tag{2.1}$$

where $\mathbf{n} \in \mathbb{R}^m$ denotes noise that is typically assumed Gaussian. Traditionally, (2.1) has been studied under the assumption of $\mathbf{x}$ belonging to a low-dimensional subspace of $\mathbb{R}^m$ [27, 28, 29, 30]. In contrast, our focus is on the case of $\mathbf{x}$ belonging to a *union* of low-dimensional subspaces: $\mathbf{x} \in \bigcup_{k=1}^{K_0} S_k$, where $S_k \subset \mathbb{R}^m$ denotes a subspace of $\mathbb{R}^m$. We further assume that the subspaces are pairwise disjoint, $S_k \cap S_{k'} = \emptyset$ for $k \neq k'$, and they have the same dimension: $\forall k, \dim(S_k) = n \ll m$.[1]

The second problem studied in this chapter, which does not arise in classical subspace detection literature, is referred to as *active subspace detection*. The goal in this problem is

[1] One can extend this work to the case of different dimensional subspaces in a straightforward manner at the expense of notational complexity.

to not only detect whether $\mathbf{y}$ contains an unknown signal $\mathbf{x}$, but also *identify* the subspace S_k to which $\mathbf{x}$ belongs. Mathematically, this can be posed as a multiple hypothesis test with the null ($\mathcal{H}_0$) and the alternate ($\{\mathcal{H}_k\}_{k=1}^{K_0}$) hypotheses given by:

$$\begin{aligned} \mathcal{H}_0 : &\quad \mathbf{y} = \mathbf{n}; \\ \mathcal{H}_k : &\quad \mathbf{y} = \mathbf{x} + \mathbf{n},\ \mathbf{x} \in S_k; \quad k = 1, \ldots, K_0. \end{aligned} \tag{2.2}$$

Our goal in this chapter is to derive statistical tests for (2.1) and (2.2), and provide a rigorous mathematical understanding of the performance of the derived tests. Our analysis is based on the assumption of $\mathbf{n}$ being a colored Gaussian noise that is distributed as $\mathcal{N}(0, \sigma^2\mathbf{R})$ with $\mathbf{R}$ being a full-rank covariance. In particular, we focus on the three cases of (i) known noise statistics, (ii) known variance (σ^2), but unknown covariance ($\mathbf{R}$), and (iii) unknown variance and covariance. In contrast to prior works [1, 2, 3, 24, 26], we are specifically interested in characterizing our results in terms of the geometry of the underlying subspaces. This geometry can be described through the principal angles between the subspaces, where the i-th principal angle between subspace S_j and S_k, denoted by $\varphi_i^{(j,k)}, i = 1, \ldots, n$, is recursively defined as [33]:

$$\begin{aligned} \varphi_i^{(j,k)} = \arccos\Bigg(\max_{\mathbf{u},\mathbf{v}} \Bigg\{ \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2} : \mathbf{u} \in S_j, \mathbf{v} \in S_k, \\ \mathbf{u} \perp \mathbf{u}_\ell, \mathbf{v} \perp \mathbf{v}_\ell, \ell = 1, \ldots, i-1 \Bigg\}\Bigg), \end{aligned} \tag{2.3}$$

where $(\mathbf{u}_\ell, \mathbf{v}_\ell) \in S_j \times S_k$ denote the principal vectors associated with the ℓ-th principal angle. It is straightforward to see that $0 \leq \varphi_1^{(j,k)} \leq \varphi_2^{(j,k)} \leq \cdots \leq \varphi_n^{(j,k)} \leq \pi/2$.

We conclude by noting that our statistical tests in the following will be expressed in

terms of the following ratios for compactness purposes:

$$T_{\mathbf{z}}(\mathbf{P}) = \frac{\mathbf{z}^T\mathbf{P}\mathbf{z}}{\mathbf{z}^T\mathbf{z}},\ T_{\mathbf{z}}^{\eta}(\mathbf{P}) = \frac{\mathbf{z}^T\mathbf{P}\mathbf{z}}{\eta},$$
$$T_{\mathbf{z}}(\mathbf{P},\mathbf{Q}) = \frac{\mathbf{z}^T\mathbf{P}\mathbf{z}}{\mathbf{z}^T\mathbf{Q}\mathbf{z}},\ \overline{T}_{\mathbf{z}}^{\eta}(\mathbf{P}) = \frac{\mathbf{z}^T\mathbf{P}\mathbf{z}}{\eta + \mathbf{z}^T\mathbf{z}},$$

where $\mathbf{z}$ and $(\mathbf{P}, \mathbf{Q})$ denote a vector and matrices of appropriate dimensions, respectively, while $\eta > 0$ denotes a constant.

2.2.1 Performance metrics

The performances of the statistical tests proposed in this chapter will be characterized in terms of the probabilities of detection (P_D), classification (P_C), and false alarm (P_{FA}). Specifically, let $P_{\mathcal{H}_i}(\cdot) = \Pr(\cdot|\mathcal{H}_i)$, and define the event $\widehat{\mathcal{H}}_i = \{\text{Hypothesis } \mathcal{H}_i \text{ is accepted}\}$. Then, in the case of signal detection, we have $P_D = P_{\mathcal{H}_1}(\widehat{\mathcal{H}}_1)$ and $P_{FA} = P_{\mathcal{H}_0}(\widehat{\mathcal{H}}_1)$. In contrast, in the case of active subspace detection, we have $P_C = \sum_{k=1}^{K_0} P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k)\Pr(\mathcal{H}_k)$ and $P_{FA} = P_{\mathcal{H}_0}(\cup_{k=1}^{K_0}\widehat{\mathcal{H}}_k)$.

We conclude by pointing out that some of our forthcoming discussion will use the shorthand $P_{S_k}(\cdot) = \Pr(\cdot|\{\mathbf{x} \in S_k\})$ and $\Psi(\eta_0, \alpha) = \frac{\sqrt{2}}{2^n\Gamma(n/2)}(\eta_0\alpha)^{(n-1)/2}K_{(n-1)/2}\left(\frac{\eta_0\alpha}{2}\right)$, where $\alpha \in \mathbb{R}_+$ and $\eta_0 \in (0, 1/2)$. Using this notation, we can also write the probability of detection as $P_D = \sum_{k=1}^{K_0} P_{S_k}(\widehat{\mathcal{H}}_1)\Pr(\mathbf{x} \in S_k)$.

2.3 Main results

In this section, we present statistical tests for both the detection problems under various noise conditions. In addition, we provide bounds on the performance metrics for these tests.

2.3.1 Known noise statistics

We begin with the assumption that both the noise variance, σ^2, and the covariance, $\mathbf{R}$, are known. It is trivial to see that $\mathbf{y}|\mathcal{H}_0 \sim \mathcal{N}(0, \sigma^2\mathbf{R})$ for both detection problems. Further, in the case of signal detection, we have $\mathbf{y}|\mathcal{H}_1 \sim \mathcal{N}(\mathbf{x}, \sigma^2\mathbf{R})$. In contrast, the observations $\mathbf{y}$ under the k-th alternate hypothesis in the case of active subspace detection can be expressed as $\mathbf{y}|\mathcal{H}_k \sim \mathcal{N}(\mathbf{H}_k\boldsymbol{\theta}_k, \sigma^2\mathbf{R}), k = 1, \ldots, K_0$, where $\mathbf{H}_k \in \mathbb{R}^{m\times n}$ denotes a basis for subspace S_k and $\boldsymbol{\theta}_k \in \mathbb{R}^n$ denotes representation coefficients of $\mathbf{x}$ under basis $\mathbf{H}_k$. Since $\mathbf{x}$ and $\boldsymbol{\theta}_k$ are unknown for the signal and the active subspace detection problems, respectively, we resort to the *generalized likelihood ratio tests* (GLRTs) for the two detection problems. Our results in this regard are based on the following definitions: let $\mathbf{z} = \mathbf{R}^{-\frac{1}{2}}\mathbf{y}$ denote the *whitened* observations, $\mathbf{w} = \mathbf{R}^{-\frac{1}{2}}\mathbf{n}$ denote the *whitened* noise, $\mathbf{G}_k = \mathbf{R}^{-\frac{1}{2}}\mathbf{H}_k, k = 1, \ldots, K_0$, denote the whitened subspace bases, and $\mathbf{P}_{\bar{S}_k} = \mathbf{G}_k(\mathbf{G}_k^T\mathbf{G}_k)^{-1}\mathbf{G}_k^T$ and $\mathbf{P}_{\bar{S}_k}^{\perp} = \mathbf{I} - \mathbf{P}_{\bar{S}_k}$, respectively, denote the projection matrix for the k-th whitened subspace and its orthogonal complement.

Theorem 1. *Let $\bar{\gamma} > 0$ denote the test threshold and define $\widehat{k} = \arg\max_k(\mathbf{z}^T\mathbf{P}_{\bar{S}_k}\mathbf{z})$. The GLRT for the signal detection and the active subspace detection problem is, respectively, given by*

$$T_{\mathbf{z}}^{2\sigma^2}\left(\mathbf{P}_{\bar{S}_{\widehat{k}}}\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma} \qquad \textit{and} \qquad T_{\mathbf{z}}^{2\sigma^2}\left(\mathbf{P}_{\bar{S}_{\widehat{k}}}\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \bar{\gamma}\,. \tag{2.4}$$

The proof of this theorem is given in Appendix 2.7.1, while its interpretation as well as its relationship to the classical test for subspace detection are provided in Sec. 2.4. We now characterize the performance of the statistical tests in (2.4) in terms of bounds on P_{FA}, P_D, and P_C. Note that we have to resort to bounds, as opposed to exact expressions, because of the complicated joint distributions that arise in our context; we refer the reader to Appendix 2.7.2 for further discussion.

Theorem 2. *The GLRTs in Theorem 1 for the signal and the active subspace detection problems result in probability of false alarm that is upper bounded by:*

$$P_{FA} \leq \min\Big\{1\ ,\ \sum_{k=1}^{K_0} \Pr\Big(T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\Big)\Big\}. \tag{2.5}$$

Further, in the case of signal detection, the probability of detection $P_D = \sum_{k=1}^{K_0} P_{S_k}(\widehat{\mathcal{H}}_1)\Pr(\mathbf{x} \in S_k)$ *can be upper and lower bounded by the fact that*

$$\begin{aligned} P_{S_k}(\widehat{\mathcal{H}}_1) &\leq \min\Big\{1\ ,\ \sum_{i=1}^{K_0} P_{S_k}\Big(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}\Big)\Big\},\ \textit{and} \\ P_{S_k}(\widehat{\mathcal{H}}_1) &\geq \sum_{i=1}^{K_0} \frac{\Big[P_{S_k}\Big(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}\Big)\Big]^2}{\sum_{j=1}^{K_0} P_{S_k}\Big(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}, T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_j}) > \bar{\gamma}\Big)}. \end{aligned} \tag{2.6}$$

Finally, the probability of classification P_C *for active subspace detection can be lower bounded by the fact that*

$$\begin{aligned} P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) \geq \max\Big\{0\ ,\ & P_{S_k}(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}) + \\ & \sum_{j=1, j\neq k}^{K_0} P_{S_k}(T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1) - (K_0 - 1)\Big\}. \end{aligned} \tag{2.7}$$

The proof of this theorem is given in Appendix 2.7.2. It is worth noting that probabilities of the form $P_{S_k}(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_j}) > \bar{\gamma})$ correspond to tail probabilities of chi-squared random variables, whereas the probabilities $P_{S_k}(T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1)$ involve ratios of *dependent* chi-squared variables whose distributions can be numerically computed.

Remark 1. It is noted in Appendix 2.7.2 that (2.7) can be further lower bounded using [26, Lemma 1] as $P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) \geq \max\big\{0, P_{S_k}(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}) - \sum_{j:j\neq k} Q\big(\frac{1}{2}(1-2\eta_0)\sqrt{\lambda_{j\backslash k}}\big) - \sum_{j:j\neq k} \Psi(\eta_0, \lambda_{j\backslash k})\big\}$, where $\lambda_{j\backslash k} = \mathbf{z}^T \mathbf{P}_{\bar{S}_j}^{\perp} \mathbf{z}/\sigma^2$ when $\mathbf{z} \in \bar{S}_k$. This bound, however, depends further on η_0. Numerical experiments reported in Sec. 2.5 show the looseness of this bound for the case of $\eta_0 = 0.25$, the value advertised in [26].

Remark 2. A heuristic approach to detecting signals under the UoS model would be to use the multiple hypothesis tests of [23], where each test is an individual matched subspace detector. The final decision can then be made by taking the union of binary outputs from each matched detector and declaring detection if any one of them has detected a signal. It is straightforward to see however that this final decision rule coincides with the decision rule in (2.4). Thus, in the event that only one subspace is active, the testing procedure in [23] effectively reduces to a GLRT.

2.3.2 Unknown noise covariance

Next, we consider the case of colored noise with unknown covariance matrix $\mathbf{R}$. In this case, we also assume access to N_0 noise samples $\boldsymbol{\xi}_p \sim \mathcal{N}(0, \mathbf{R}), p = 1, \ldots, N_0$ ($N_0 > m$ to obtain a non-singular estimate of $\mathbf{R}$), which is a standard assumption in the detection literature [28, 29, 30]. As before, we use GLRTs to obtain decision rules for the two detection problems. Our results make use of the following definitions: let $\boldsymbol{\Sigma} = \frac{1}{N_0}\sum_{p=1}^{N_0} \boldsymbol{\xi}_p \boldsymbol{\xi}_p{}^T$ denote sample covariance of noise samples, $\widehat{\mathbf{z}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{y}$ denote the *empirically whitened* observations, $\widehat{\mathbf{w}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{n}$ denote the *empirically whitened* noise, $\widehat{\mathbf{G}}_k = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{H}_k, k = 1, \ldots, K_0$, denote the empirically whitened subspace bases, and $\widehat{\mathbf{P}}_{\bar{S}_k} = \widehat{\mathbf{G}}_k(\widehat{\mathbf{G}}_k^T\widehat{\mathbf{G}}_k)^{-1}\widehat{\mathbf{G}}_k^T$ denote the projection matrix for the k-th empirically whitened subspace.

Theorem 3. *Let $\bar{\gamma} > 0$ denote the test threshold and define $\widehat{k} = \arg\max_k(\widehat{\mathbf{z}}^T\widehat{\mathbf{P}}_{\bar{S}_k}\widehat{\mathbf{z}})$. The GLRT for the signal detection and the active subspace detection problem is, respectively, given by:*

$$\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma} \qquad \textit{and} \qquad \overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \bar{\gamma}. \tag{2.8}$$

The proof of this theorem is provided in Appendix 2.7.3, while some discussion on interpretation and relationship to the classical test for subspace detection is provided in Sec. 2.4. We now characterize the performance of the statistical tests in (2.8) in terms of

bounds on P_{FA}, P_D, and P_C.

Theorem 4. *The GLRTs for the signal and the active subspace detection problems in Theorem 3 result in the probability of false alarm that is upper bounded by:*

$$P_{FA} \leq \min\Big\{1\ ,\ \sum_{k=1}^{K_0} \Pr\Big(\overline{T}_{\widehat{\mathbf{w}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_k}) > \bar{\gamma}\Big)\Big\}. \tag{2.9}$$

Further, the detection probability for signal detection, $P_D = \sum_{k=1}^{K_0} P_{S_k}(\widehat{\mathcal{H}}_1)\Pr(\mathbf{x} \in S_k)$ *can be upper and lower bounded by the fact that*

$$\begin{aligned} P_{S_k}\Big(\widehat{\mathcal{H}}_1\Big) &\leq \min\Big\{1\ ,\ \sum_{i=1}^{K_0} P_{S_k}\Big(\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_i}) > \bar{\gamma}\Big)\Big\},\ \textit{and} \\ P_{S_k}\Big(\widehat{\mathcal{H}}_1\Big) &\geq \sum_{i=1}^{K_0} \frac{\Big[P_{S_k}\Big(\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_i}) > \bar{\gamma}\Big)\Big]^2}{\sum_{j=1}^{K_0} P_{S_k}\Big(\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_i}) > \bar{\gamma}, \overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_j}) > \bar{\gamma}\Big)}. \end{aligned} \tag{2.10}$$

Finally, the probability of classification P_C *for active subspace detection can be lower bounded by the fact that*

$$\begin{aligned} P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) \geq \max\Big\{0\ ,\ & P_{S_k}(\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_k}) > \bar{\gamma}) + \\ & \sum_{j=1, j\neq k}^{K_0} P_{S_k}(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_k}, \widehat{\mathbf{P}}_{\bar{S}_j}) > 1) - (K_0 - 1)\Big\}. \end{aligned} \tag{2.11}$$

The proof of this theorem follows along similar lines as for the proof of Theorem 2 and is thus omitted. In contrast to Theorem 2, the terms of the form $P_{S_k}(\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_k}) > \bar{\gamma})$ and $P_{S_k}(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_k}, \widehat{\mathbf{P}}_{\bar{S}_j}) > 1)$ involve probabilities of the ratios of *dependent* chi-squared variables and have to be computed numerically.

Remark 3. One can again further lower bound (2.11) using [26, Lemma 1] as: $P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) \geq \max\Big\{0, P_{S_k}(\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_k}) > \bar{\gamma}) - \sum_{j:j\neq k}^{K_0} Q\big(\frac{1}{2}(1-2\eta_0)\sqrt{\widehat{\lambda}_{j\backslash k}}\big) - \sum_{j:j\neq k}^{K_0} \Psi(\eta_0, \widehat{\lambda}_{j\backslash k})\Big\}$, where $\widehat{\lambda}_{j\backslash k} = \frac{1}{\sigma^2}\widehat{\mathbf{z}}^T\widehat{\mathbf{P}}_{\bar{S}_j}^{\perp}\widehat{\mathbf{z}}$ when $\mathbf{z} \in \bar{S}_k$.

2.3.3 Unknown noise statistics

We now address adaptive detection in settings where the covariance matrix $\mathbf{R}$ and variance σ^2 are both unknown. Once again assuming access to N_0 noise samples and using the notation of Sec. 2.3.2, the GLRTs lead to the following decision rules.

Theorem 5. *Let $\bar{\gamma} > 0$ denote the test threshold and define $\widehat{k} = \arg\max_k(\widehat{\mathbf{z}}^T \widehat{\mathbf{P}}_{\bar{S}_k} \widehat{\mathbf{z}})$. The GLRT for the signal detection and the active subspace detection problem is, respectively, given by:*

$$T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma} \quad \textit{and} \quad T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \bar{\gamma}. \tag{2.12}$$

The proof of this theorem is given in Appendix 2.7.4, with corresponding discussion in Sec. 2.4. The performance of the statistical tests in (2.12) is given by the following theorem.

Theorem 6. *The GLRTs for the signal and the active subspace detection problems in Theorem 5 result in the probability of false alarm that is upper bounded by:*

$$P_{FA} \leq \min\Big\{1\ ,\ \sum_{k=1}^{K_0} \Pr\Big(T_{\widehat{\mathbf{w}}}(\widehat{\mathbf{P}}_{\bar{S}_k}) > \bar{\gamma}\Big)\Big\}. \tag{2.13}$$

Further, signal detection probability $P_D = \sum_{k=1}^{K_0} P_{S_k}(\widehat{\mathcal{H}}_1)\Pr(\mathbf{x} \in S_k)$ can be upper and lower bounded by the fact that

$$\begin{aligned} P_{S_k}\Big(\widehat{\mathcal{H}}_1\Big) &\leq \min\Big\{1\ ,\ \sum_{i=1}^{K_0} P_{S_k}\Big(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_i}) > \bar{\gamma}\Big)\Big\},\ \textit{and} \\ P_{S_k}\Big(\widehat{\mathcal{H}}_1\Big) &\geq \sum_{i=1}^{K_0} \frac{\Big[P_{S_k}\Big(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_i}) > \bar{\gamma}\Big)\Big]^2}{\sum_{j=1}^{K_0} P_{S_k}\Big(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_i}) > \bar{\gamma}, T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_j}) > \bar{\gamma}\Big)}. \end{aligned} \tag{2.14}$$

Finally, the probability of classification P_C for active subspace detection can be lower

bounded by the fact that

$$P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) \geq \max\Big\{0\ ,\ P_{S_k}(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_k}) > \bar{\gamma}) + \sum_{j=1, j\neq k}^{K_0} P_{S_k}(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_k}, \widehat{\mathbf{P}}_{\bar{S}_j}) > 1) - (K_0 - 1)\Big\}. \tag{2.15}$$

The proof of this theorem is also similar to the proof of Theorem 2 and is thus omitted. Similar to Theorem 4, the terms of the form $P_{S_k}(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_k}) > \bar{\gamma})$ and $P_{S_k}(T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_k}, \widehat{\mathbf{P}}_{\bar{S}_j}) > 1)$ need to be computed numerically.

Remark 4. Similar to Remark 3, a looser lower bound can be derived here as well, with the only difference being that $\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_k})$ is replaced by $T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_k})$.

2.4 Discussion

In this section we discuss some characteristics of the various test statistics obtained in Sec. 2.3. We also describe the impact of geometry of the subspaces in the union and the geometry of the colored noise on the detection performances.

2.4.1 UoS detection versus classical subspace detection

First, we compare the test statistics for signal detection under the UoS model ((2.4),(2.8) and (2.12)) with their counterparts under the subspace model [27, 28, 29, 30]. Under the subspace observation model, the signal $\mathbf{x}$ is assumed to belong to a single subspace, $\mathbf{x} = \mathbf{H}\boldsymbol{\theta}$, where $\mathbf{H}$ contains the subspace bases. The corresponding test statistics for known noise statistics, unknown noise covariance and unknown noise statistics are, respectively, given by [27, 28, 29, 30]:

$$T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma}, \overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma}, \text{ and } T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma}. \tag{2.16}$$

At a first glance, the test statistics for the UoS model and the subspace model look similar. However, the numerator of the statistics for the subspace model corresponds to the energy of the observed signal after projection onto the relevant subspace. In contrast, since we deal with multiple subspaces, we have to rely on projection onto the subspace that captures the most energy of the observed signal.

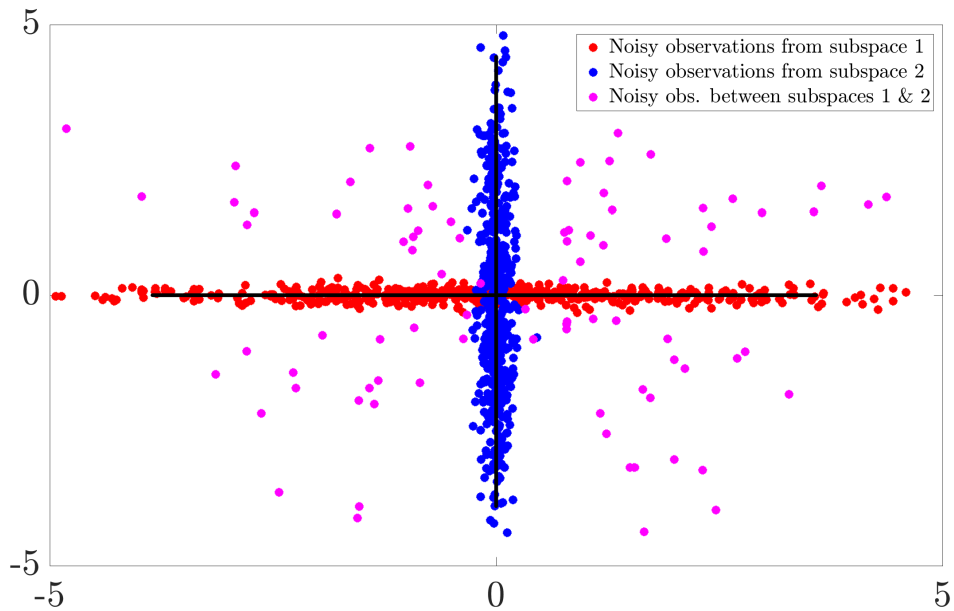

Figure 2.1: This figure highlights the difference between UoS- and classical subspace-based detection of signals generated under the UoS model. The red and blue dots correspond to noisy signals generated from a union of two subspaces, while the magenta dots represent observations that do not belong to the union. UoS-based detection would be able to reject the magenta observations, whereas subspace-based detection would accept them as signals since they belong to the direct sum of the underlying subspaces.

Next, we discuss advantages of the UoS-based test statistics over the respective subspace-based test statistics for signal detection. Under the assumption of the (noisy) signal being generated under the UoS model, the test statistics derived in this chapter reject observations that correspond to the "gaps" between the individual subspaces; see, e.g., Fig. 2.1, in which observations in the gaps correspond to magenta-colored dots. In contrast, subspace-based detection needs to resort to direct sum of the underlying subspaces in the union. This, in turn, leads to higher false alarm rates since observations in gaps that belong to the direct

sum are falsely accepted as signals; in Fig. 2.1, e.g., subspace-based detection will accept all magenta observations as signals. We also refer the reader to Sec. 2.5 for numerical validation of this fact.

Finally, the presence of multiple subspaces in the union also results in the problem of active subspace detection, which does not arise in the context of classical subspace detection as it only considers one underlying subspace.

2.4.2 Signal detection versus active subspace detection

Notice that the test statistics for active subspace detection have forms similar to those for signal detection. The main difference lies in the performance of these statistics when detecting either the signal or the active subspace. The detection performance for active subspace detection is lower than that for signal detection. This is due to the fact that for signal detection, the detector is not concerned with detecting the true subspace from which the observed signal is coming and can afford to confuse one subspace with another as long as it detects the presence of a signal. That is not the case with active subspace detection, where this confusion matters, and thus we observe the loss in performance. This fact was also highlighted by Gini et al. in [24].

2.4.3 Invariance properties of the test statistics

We now examine the invariance properties of our test statistics for signal detection. Since our test statistics for active subspace detection are similar to those for signal detection under UoS model, they exhibit similar invariance properties.

From the expressions in (2.4), (2.8) and (2.12), notice that the statistics are invariant to the rotations in $\bar{S}_{\widehat{k}}$. This means all rotated versions of the relevant signal (for rotations in $\bar{S}_{\widehat{k}}$) will result in same detection performance. Moreover, the statistics also exhibit invariance with respect to the translations in $\bar{S}_{\widehat{k}}^{\perp}$ (which is the orthogonal subspace of $\bar{S}_{\widehat{k}}$). This implies that any additive interference from $\bar{S}_{\widehat{k}}^{\perp}$ is unnoticeable to the detectors since they only

measure the energy of $\mathbf{z}$ in the subspace $\bar{S}_{\widehat{k}}$. Additionally, the test statistic for detection in unknown noise statistics (2.12) is also invariant to the scaling of the observed signals, i.e., scaled versions of a signal will result in same detection performance with this test. This is due to the fact that both numerator and denominator in (2.12) are quadratic forms of the *whitended/empirically whitened* observations $\mathbf{z}$, without any additive terms.

2.4.4 Influence of geometry between whitened subspaces on detection probability

The detection performance of our detector decreases only slightly as the angles between the subspaces increase. This can be seen from an alternate expression for the probability of union of events. For example, the probability of union of two events, A, and B, can be written as: $P(A \cup B) = P(A) + P(B) - P(AB) = P(AB') + P(A'B) + P(AB)$ where A', and B' are the complements of the corresponding events. One can thus see that the probability of union of events is directly proportional to the probability of the intersection of events (and their complements). For the case of detection probability, these intersections are k-tuples of the form $\bigcap_{j=1}^{k} \left\{ T_{\widehat{\mathbf{z}}}^{2\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_j}) > \bar{\gamma} \right\}$ (and their complements). When a pair (or more) of subspaces are close to each other, i.e., the principal angles between whitened/empirically whitened subspaces are small, the probability of these k-tuples is larger compared to when the subspaces are far apart.

Intuitively, since signal detection problem is not concerned with the detection of the active subspace, confusing a (noisy) signal coming from one subspace as being generated from another subspace does not matter significantly. In fact, this confusion helps the detection task as long as a signal is actually present. Interestingly, when the subspaces are far apart, i.e., principal angles are large, chances of such confusion are less and the probability of detection is slightly decreased.

2.4.5 Influence of geometry between whitened subspaces on correct classification probability

We now examine the influence of geometry between whitened subspaces on the probability of correct classification. This analysis in particular sets us apart from other related works such as [1, 2, 3, 24, 26], as we make the influence of geometry explicit through the principal angles between subspaces. We start with the case of active subspace detection in known noise statistics. The crux of our analysis is given in the following theorem.

Theorem 7. *When the active subspaces are detected using the test in Theorem 1, the lower bound on the probability of correct classification increases with increasing principal angles between the whitened subspaces.*

The proof of this theorem is detailed in Appendix 2.7.5. The following corollary can also be obtained form Theorem 7.

Corollary 1. *Suppose the noise is white Gaussian, i.e.,* $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. *When the active subspaces are detected using the test in Theorem 1, the lower bound on the probability of correct classification increases with increasing principal angles between the subspaces in the union.*

Similarly, in the case of other noise settings (unknown covariance and unknown noise statistics), the probability of correct classification of individual subspaces increases with increasing principal angles between the *empirically whitened subspaces*. This also follows trivially from Theorem 7.

2.4.6 Influence of geometry of colored noise

To characterize the effect of noise geometry on two detection problems, we focus on the terms $\mathbf{z}^T\mathbf{P}_{\bar{S}_k}\mathbf{z}$ in (2.4). We can see that $\mathbf{z}^T\mathbf{P}_{\bar{S}_k}\mathbf{z} = (\bar{\mathbf{x}} + \mathbf{w})^T\mathbf{P}_{\bar{S}_k}(\bar{\mathbf{x}} + \mathbf{w}) = \bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_k}\bar{\mathbf{x}} +$

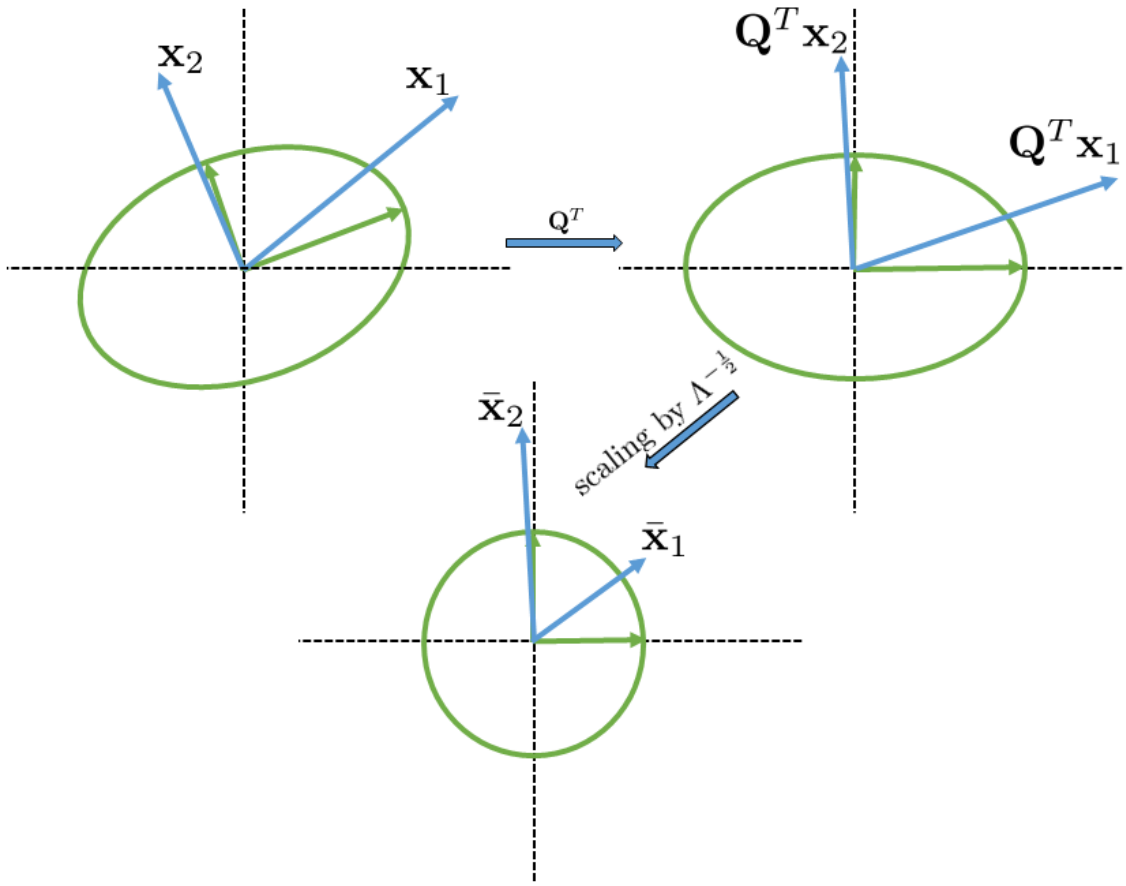

Figure 2.2: This figure shows the effect of geometry of colored noise on two signals coming from two different subspaces. The ellipse represents the covariance of the colored noise with the green vectors representing the eigenvectors of the covariance. The blue vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ represent signals from the two different subspaces. The first operation during whitening can be seen as rotation by $\mathbf{Q}^T$ to align the canonical bases with the noise eigenvectors. The second operation of scaling by $\Lambda^{-\frac{1}{2}}$ scales each axis by the inverse of the corresponding eigenvalue. Thus, the closer a subspace is to the leading eigenvectors of noise covariance, the lower is its detection probability as it suffers more attenuation during whitening.

$2\mathbf{w}^T\mathbf{P}_{\bar{S}_k}\bar{\mathbf{x}} + \mathbf{w}^T\mathbf{P}_{\bar{S}_k}\mathbf{w}$, where $\bar{\mathbf{x}} = \mathbf{R}^{-\frac{1}{2}}\mathbf{x}$. The norm of $\bar{\mathbf{x}}$ can be expressed as:

$$\|\bar{\mathbf{x}}\|_2^2 = \mathbf{x}^T\mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^T\mathbf{x} = \|\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{x}^Q\|_2^2 = \sum_{i=1}^{m}\frac{(x_i^Q)^2}{\lambda_i} \tag{2.17}$$

where $\mathbf{x}^Q = \mathbf{Q}^T\mathbf{x}$, and $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ is the eigenvalue decomposition of $\mathbf{R}$. The matrix $\mathbf{\Lambda}$ contains the eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_m$ on the diagonal and the matrix $\mathbf{Q}$ has the eigenvectors of the covariance as its columns. Note that $\mathbf{Q}^T$ is a rotation matrix that

rotates and aligns the canonical bases of the observation space with the eigenvectors of the covariance, i.e., $\mathbf{Q}^T$ performs unscaled whitening. We can see from the last expression in (2.17) that x_i^Q for smaller values of i gets attenuated by a larger λ_i than x_i^Q for larger values of i (since $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_m$). This implies that subspaces (and signals) with more energy in lower indices after unscaled whitening, suffer more attenuation and have a lower $\|\bar{\mathbf{x}}\|_2$. Thus, subspaces (signals) closer to the higher-order eigenvectors of the covariance (i.e., eigenvectors corresponding to higher eigenvalues) end up having a lower $\|\bar{\mathbf{x}}\|_2$.

With slight algebraic manipulations, we can see also that $\|\bar{\mathbf{x}}\|_2$ (and other terms proportional to it) appears in the numerator of our test statistics. This dictates that for same signal-to-noise ratio (SNR), i.e., SNR $= \frac{\|\mathbf{x}\|_2^2}{\sigma^2}$, a lower $\|\bar{\mathbf{x}}\|_2$ will result in a lower detection probability. Thus we conclude that for the same SNR, subspaces with more energy closer to the higher-order eigenvectors of the covariance have lower detection probability and vice versa. This make intuitive sense: a subspace with more influence of noise (i.e., a subspace that lives closer to the higher-order eigenvectors of the covariance) will have a lower detection rate than a subspace with less influence of noise. A depiction of this observation is shown in Fig. 2.2.

Since the same quadratic forms appear in the numerator of the test statistics for active subspace detection, we also conclude from this discussion that the subspaces with more energy near the higher-order eigenvectors of the covariance have lower probability of correct classification.

2.5 Numerical experiments

In this section, we present numerical experiments to examine the tightness of various bounds derived in this chapter and verify the trends of performance metrics with respect to the geometry of the subspaces.

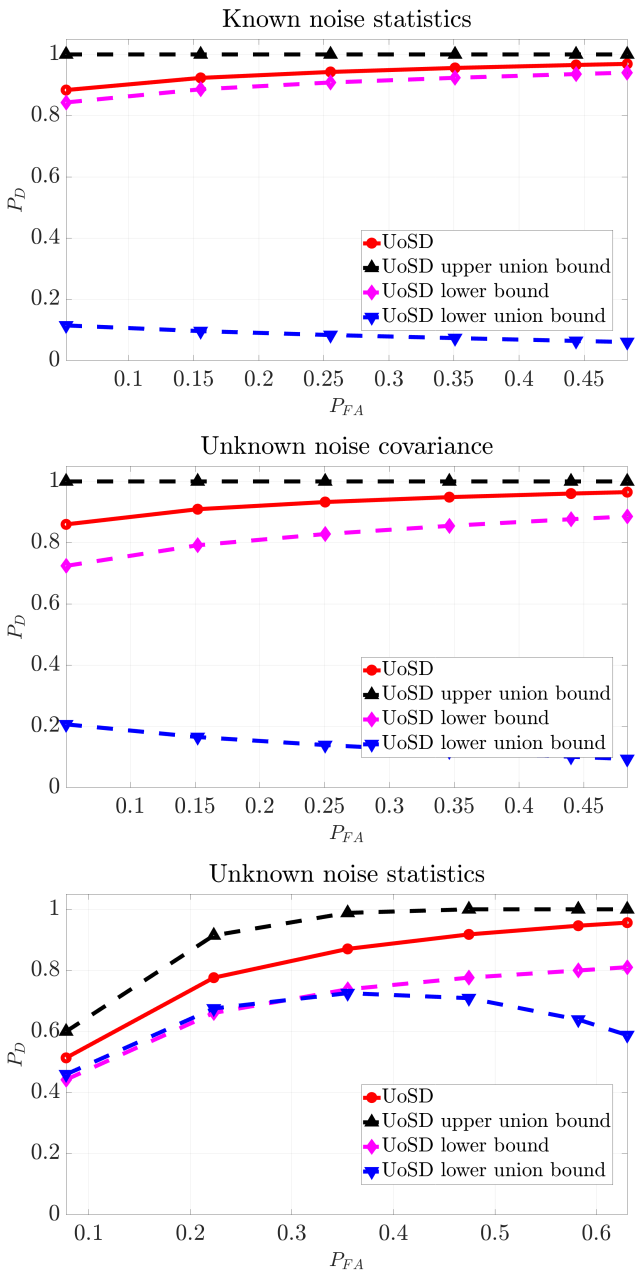

Figure 2.3: ROC curves for signal detection under the UoS model (labeled UoSD) and the derived bounds. Each subfigure shows four plots under the UoS model: the upper union bound on the detection probability, the true detection probability, the lower bound on the detection probability, and the lower union bound. Starting from the top, the subfigures show the ROC curves under known noise statistics, unknown noise covariance and unknown noise statistics, respectively.

2.5.1 Synthetic data

We run Monte-Carlo experiments for signal and active subspace detection problems under different noise settings using synthetic data. Our general procedure for these experiments is as follows: we consider a union of *three* 2-dimensional subspaces in a 4-dimensional space. The subspaces are structured to highlight the effect of geometry between subspaces. The first and third subspaces are fixed and the angles between them are kept constant. As for the second subspace, we make different realizations of it with increasing principal angles with respect to the first subspace. This process is repeated for different levels of false alarm probabilities and SNR levels. The threshold for each false alarm level is determined numerically. When unmentioned, the false alarm rate is upper bounded at 10^{-1} and the SNR is 10 dB. Each experiment is averaged over 10000 trials.

Signal detection problem

The receiver operating characteristic (ROC) curve of signal detection for the tests derived in this chapter, with their respective upper and lower bounds, is given in Fig. 2.3. We can see that the lower union bound is much looser compared to the upper union bound and the lower bound derived in the chapter. Moreover, Fig. 2.4 provides a comparison of different noise scenarios, from which we conclude that the best performance is given under known noise statistics.

Next, Fig. 2.5 shows the effect of subspace angles on the detection probability. We see that the principal angles between whitened subspaces have indeed minimal effect on the detection probability under known noise settings. A similar behavior can also be seen for detection probability under other noise settings, but we omit those plots in the interest of space.

To show the influence of the geometry of noise, we consider three 2-dimensional subspaces in a 4-dimensional space and randomly generate a noise covariance matrix. We then add noise to the eigenvectors of the noise covariance matrix and use them as bases for two

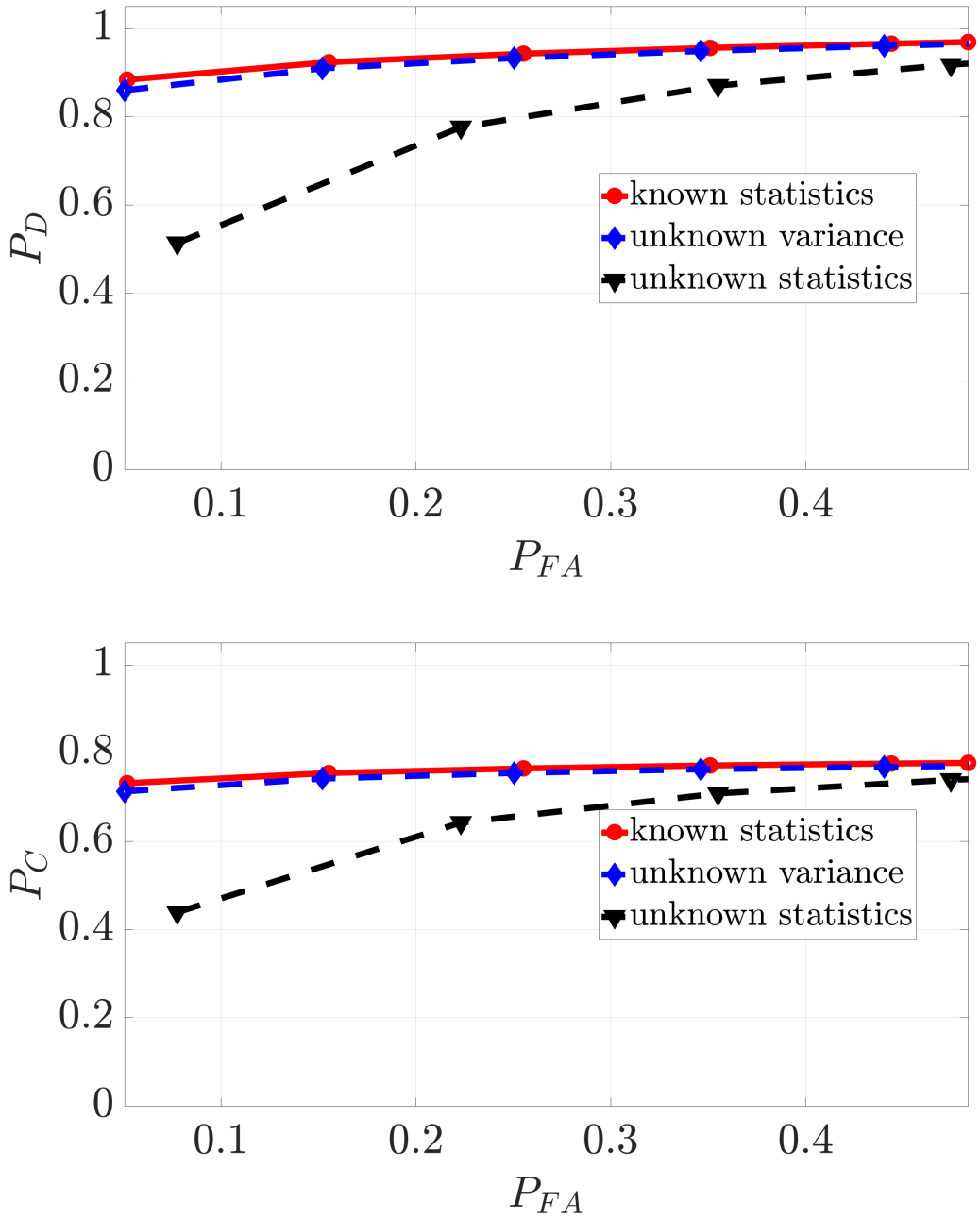

Figure 2.4: ROC curves for signal (top) and active subspace (bottom) detection under the UoS model for different noise settings.

of our subspaces. Starting from the eigenvectors corresponding to the smallest eigenvalues, we successively pick n noisy eigenvectors for subspaces S_1 and S_2 in the union. The bases of the third subspace S_3 are generated randomly from a standard normal distribution. We

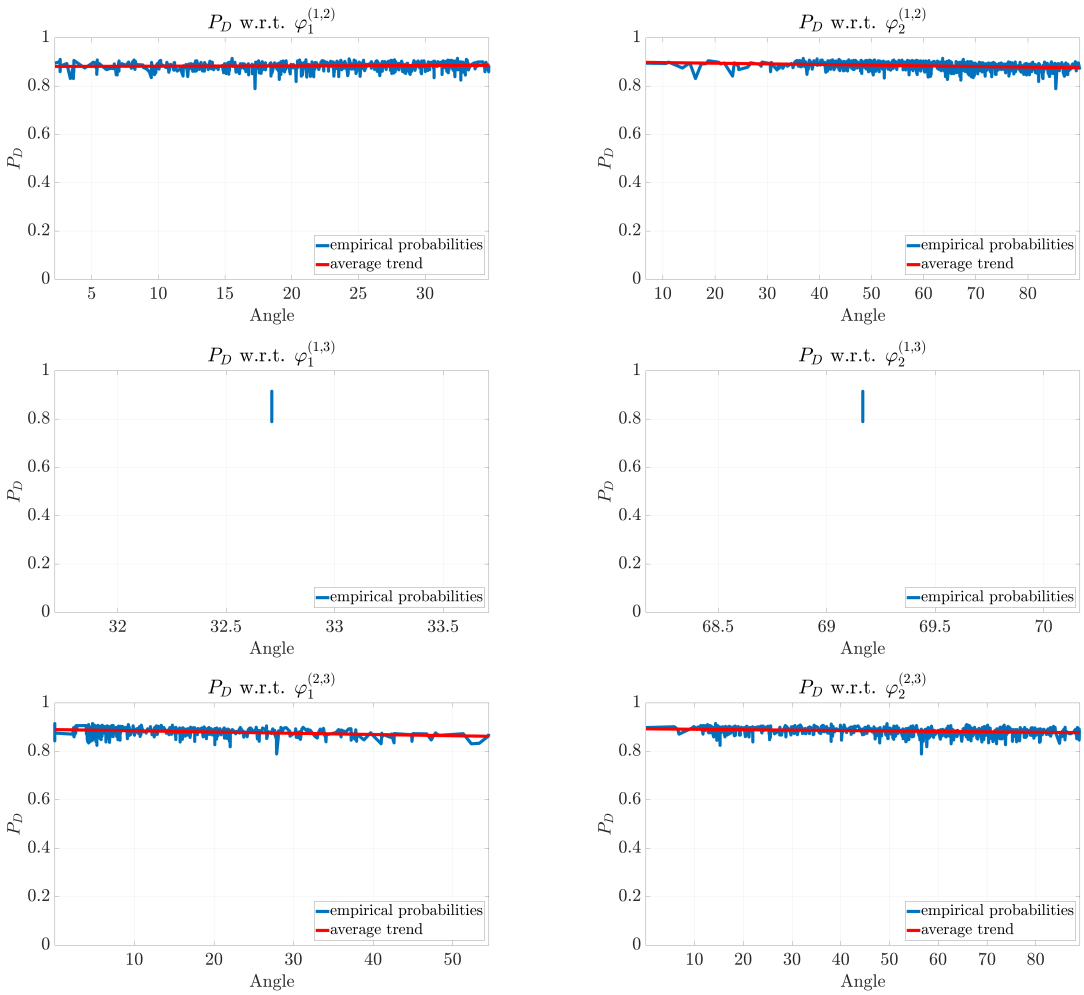

Figure 2.5: The probability of detection with respect to the principal angles between whitened subspaces when the noise statistics are fully known. The angles/whitened angles between subspaces 1 and 3 are fixed, but the probabilities change due to changing angles with subspace 2, and thus we see a vertical line for the detection probability with respect to $\varphi_1^{(1,3)}$ and $\varphi_2^{(1,3)}$. For other angles, we see a minimal decrease in probability as the angles increase.

noted in Sec. 2.4.6 that subspaces with more basis vectors closer to the higher-order eigenvectors of the noise covariance have lower $\|\bar{\mathbf{x}}\|_2$ and thus a lower detection probability, and vice versa. This trend can be clearly seen in Fig. 2.6 for signal detection under each noise setting.

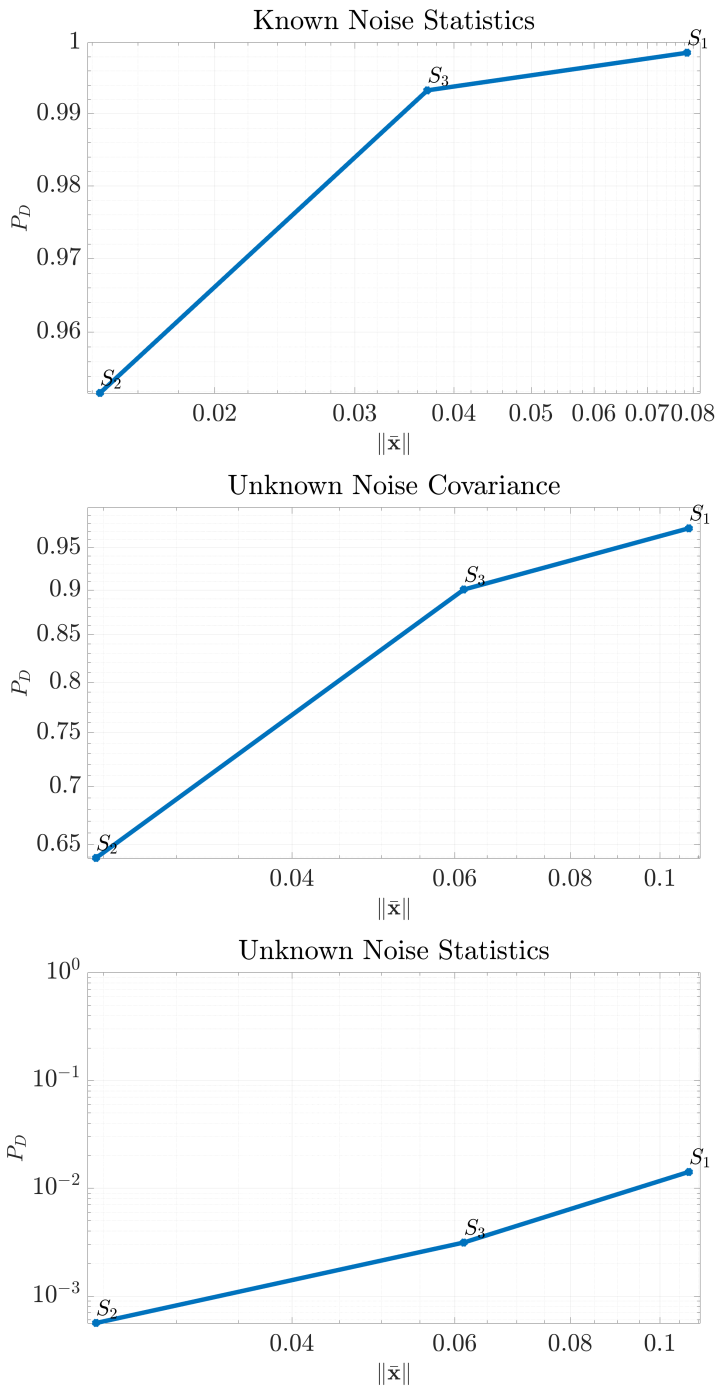

Figure 2.6: Each subfigure shows that the closer a subspace is to the higher-order eigenvectors of the noise covariance, the lower is its detection probability. On the x-axis we have the average $\|\bar{\mathbf{x}}\|$ over 12500 random signals for each subspace and the on y-axis we have the detection probability. The subspace with bases closer to the higher-order eigenvectors has lower $\|\bar{\mathbf{x}}\|$ and thus lower detection probability.

Active subspace detection problem

We now demonstrate that the probability of correct classification increases with increasing principal angles between (whitened/empirically whitened) subspaces. This trend can be seen in Fig. 2.7 for active subspace detection under known noise settings. Notice that for subspace S_2, the probability $P_{S_2}(\widehat{\mathcal{H}}_2)$ first increases then decreases. This is because as we keep increasing the angles between S_1 and S_2, S_2 keeps moving closer to S_3. Since S_1 and S_3 are fixed, the angles that S_2 collectively makes with S_1 and S_3 first increase and then decrease, resulting in the observed behavior for $P_{S_2}(\widehat{\mathcal{H}}_2)$. This insight is verified in Fig. 2.8 in terms of the plot of $\varphi_1^{1,2} + \varphi_1^{2,3}$ as a function of the number of trials. Similar trends for probabilities are seen under other noise settings, which are omitted due to space constraints.

Next, we plot the ROC curves for the probability of correct classification and the various bounds derived under different noise settings in Fig. 2.9. We see that the lower bounds derived from [26] are very loose, compared to our lower bounds. A comparison of the probability of correct classification under different noise settings is provided in Fig. 2.4.

We further show the influence of noise geometry on active subspace detection. We use the same setup as for signal detection. We can see from Fig. 2.10 that subspaces closer to the higher-order eigenvectors of the noise covariance have lower detection probability, and vice versa.

Comparison with existing approaches

Of the existing methods, we can only compare the signal detection performance of the GLRTs derived in this chapter with that of the subspace-based GLRTs. Indeed, the active subspace detection problem under the UoS model has no counterpart in the classical subspace model. Likewise, comparison with a simple GLRT (for signal detection) is also infeasible as a simple GLRT requires knowledge of the signal template, whereas we only assume access to the subspaces that generate the signal. In addition, as noted earlier in

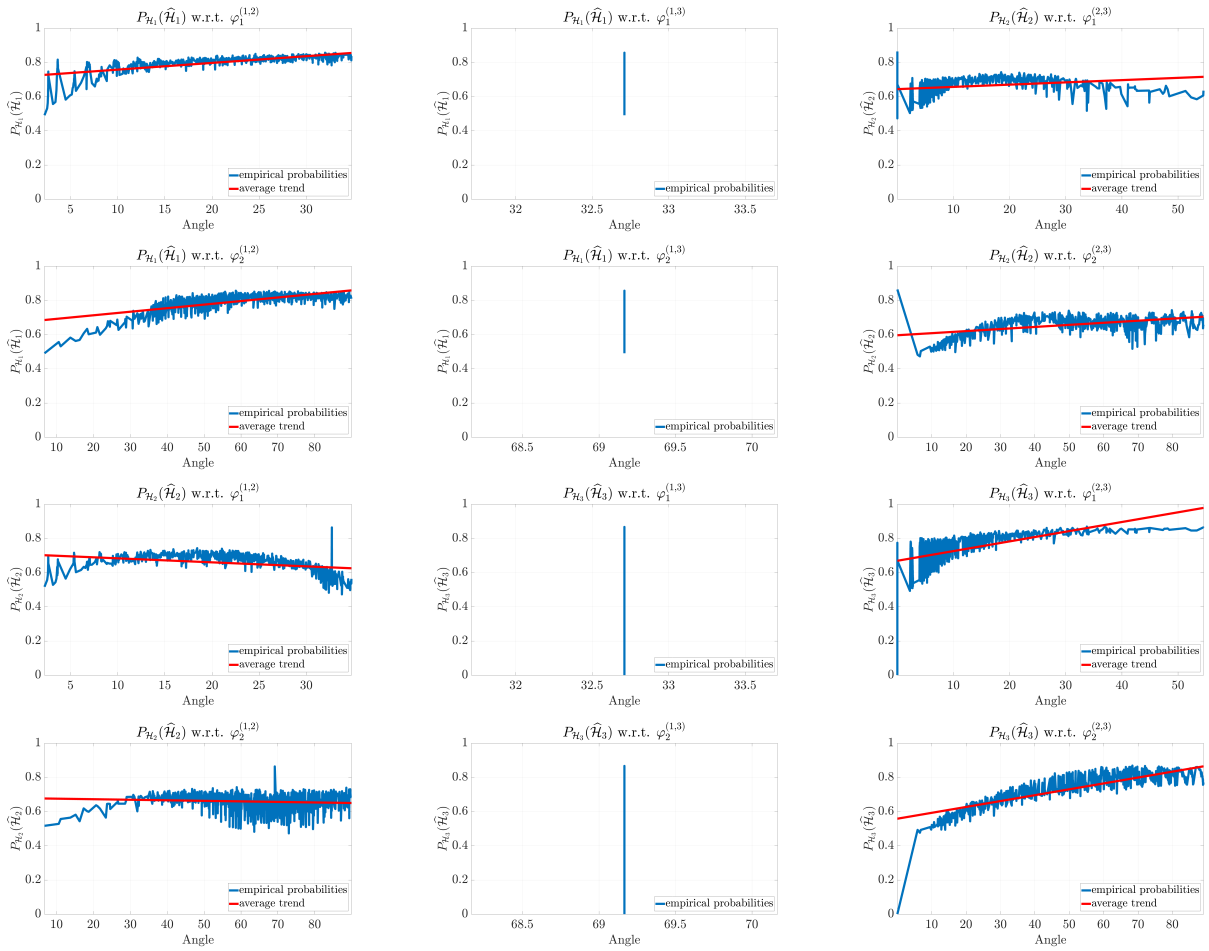

Figure 2.7: In known noise settings, the probability of correct classification increases with the increasing principal angles between whitened subspaces.

the chapter, reliance of existing compressive detection frameworks on the use of measurement matrices and (exponentially many, equiangular) random subspaces renders them impractical for UoS-based detection involving finitely many, arbitrary subspaces. In order to compare UoS-based detection with the classical subspace detection, we consider three 2-dimensional subspaces in an 8-dimensional space. Subspace detection in this setting requires projection of the observed signal onto the direct sum of the three subspaces. We compare the probability of signal detection and probability of false alarm for both UoS and subspace methods under the same SNR (5 dB) and the same detection threshold. The results, provided in Fig. 2.11 for six different threshold values, show that the probability of

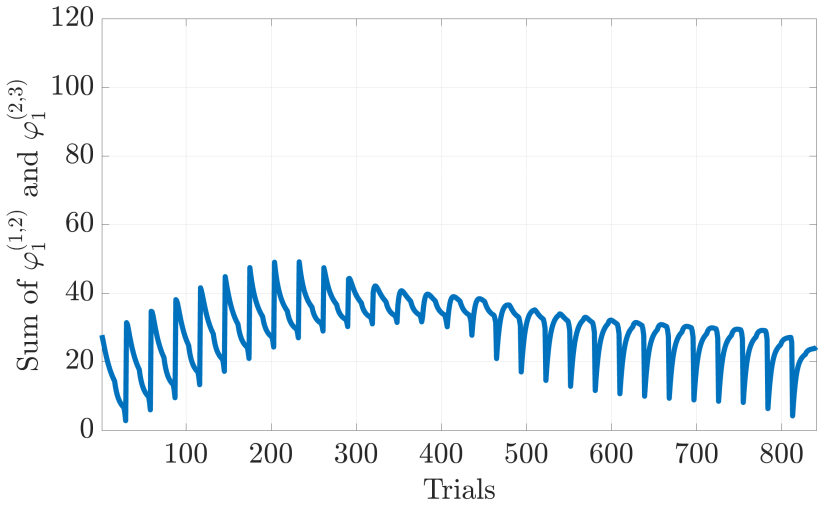

Figure 2.8: Sum of minimum principal angles subspace S_2 makes with subspace S_1 and subspace S_3. As S_2 moves away from S_1, the average of this sum increases initially and then decreases. The effect of this on the probability of classification $P_{S_2}(\widehat{\mathcal{H}}_2)$ can be seen in Fig. 2.7.

detection of the classical subspace method is slightly higher than the detection probability under the UoS model. This is because the classical subspaces model considers the direct sum of the subspaces instead of the union and ends up declaring irrelevant signals as detections. However, this in turn significantly increases the false alarm rate of signal detection under the subspace model. In particular, it can be seen from Fig. 2.11 that the probability of false alarm for the classical subspace detection far exceeds that of UoS-based detection.

Other observations

Fig. 2.12 shows the gap between detection and classification probabilities for different noise settings and different SNR levels. We can see that the gap decreases for higher SNR levels.

We make a final observation by plotting the ROC curves under various noise settings for different number of noise samples. From Fig. 2.13, we see that the gap between probabilities for known noise settings and unknown noise covariance decreases as the number

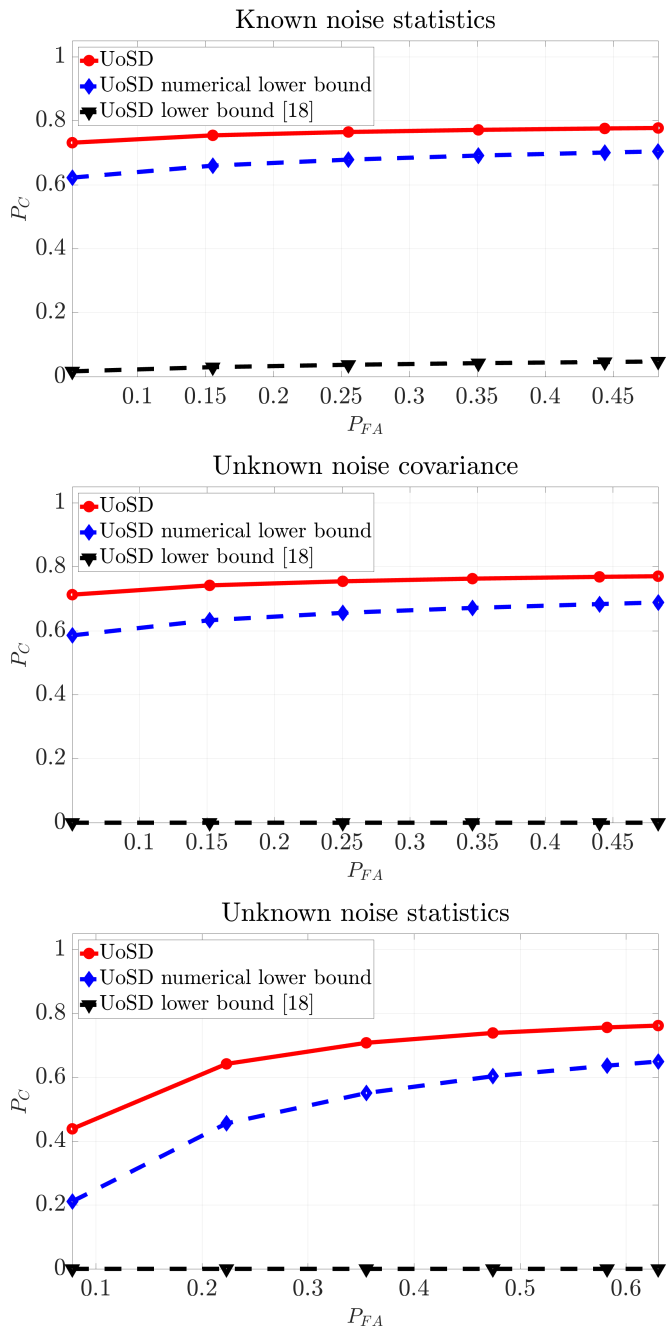

Figure 2.9: ROC curves for active subspace detection under the UoS model (labeled UoSD) and the derived bounds. All subfigures show three plots: the true classification probability under UoS, the lower bound on the classification probability computed numerically and the lower bound derived using [26]. Starting from the left, the sub-figures show the ROC curves under known noise statistics, unknown noise covariance and unknown noise statistics.

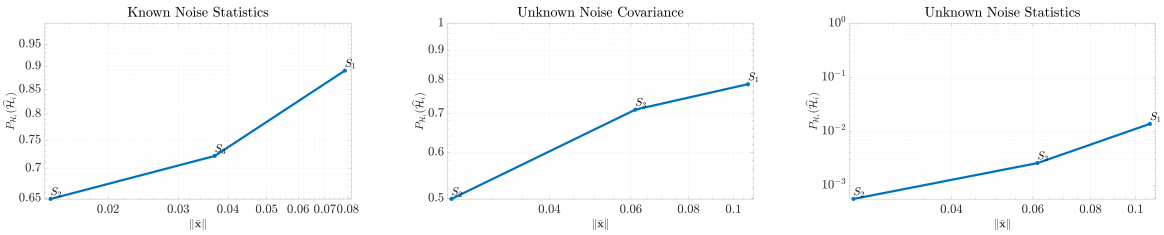

Figure 2.10: Each subfigure shows that the closer a subspace is to the higher-order eigenvectors of the noise covariance, the lower is its classification probability $P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k)$. The setup here is similar to the one for Fig. 2.6.

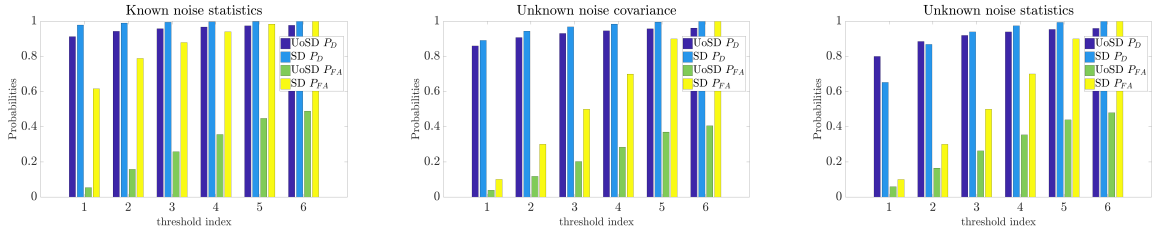

Figure 2.11: Performance comparison of UoS-based and subspace-based detection of signals generated under the UoS model. Under all noise conditions, classical subspace detection incurs a significantly higher false alarm rate than UoS-based signal detection.

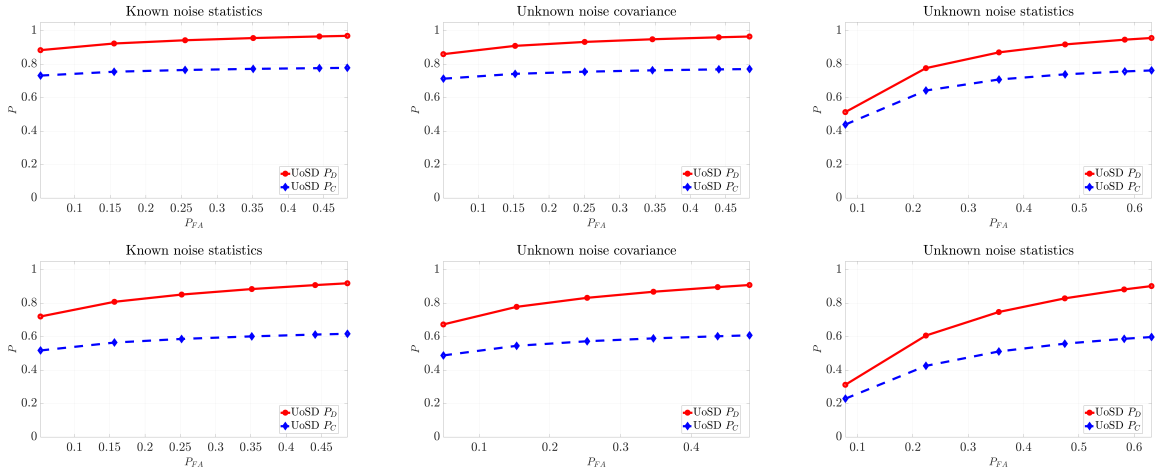

Figure 2.12: Gap between the probability of detection and the probability of correct classification under various noise settings. The two rows have SNR levels 10 dB and 5 dB, respectively. We can see that higher SNR results in a lower gap.

of noise samples increases. This is since with increasing number of noise samples, our estimates of noise statistics get better and we move closer to the regime of known noise

statistics.

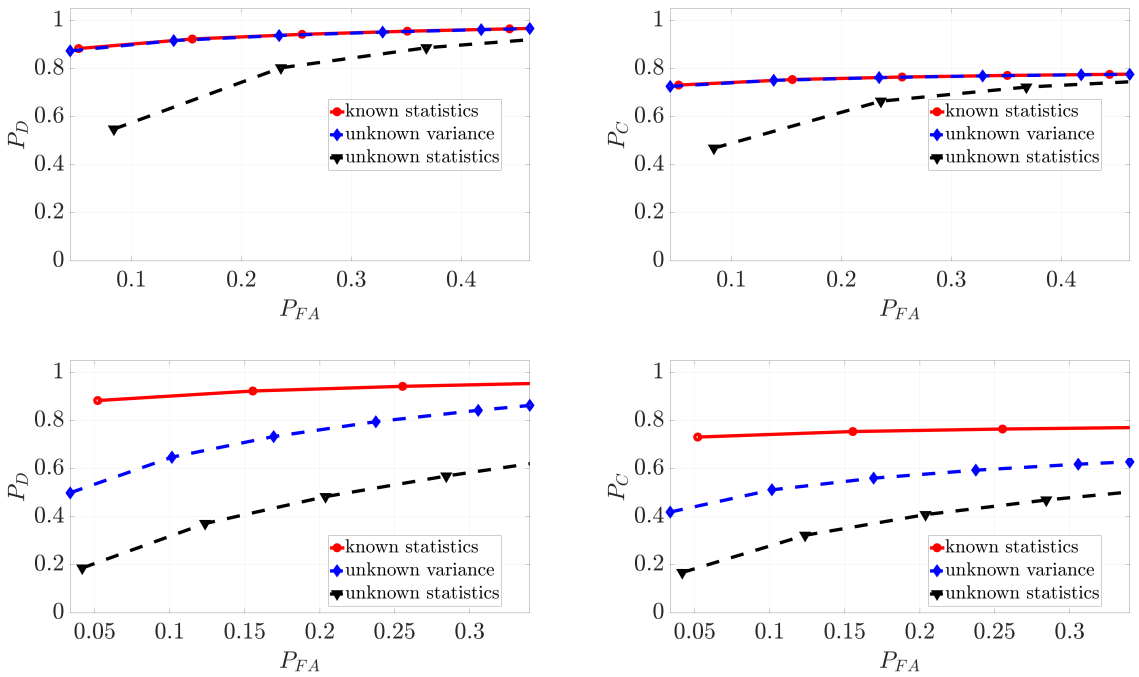

Figure 2.13: Gap between the ROC curves under various noise settings for different number of noise samples. Figures in the first row use 200 noise samples whereas the ones in the second row use 8 noise samples.

2.5.2 Real-world datasets

In this subsection, we report results on some real-world datasets that potentially conform to the UoS model. The first dataset we consider is the Salinas 'A' Scene Hyperspectral Data [34]. This data was acquired by a 224-band AVIRIS sensor over Salinas Valley (California). There are six target classes in the data. We assume each target class is lying in a different subspace, thus modeling the set of targets as belonging to a union of subspaces. To obtain the bases for the subspaces, we randomly select 20 pixels belonging to each target and use singular value decomposition (SVD) to get the bases for 10-dimensional target subsapces. For the Salinas 'A' Scene, the ground truth and the detected targets are shown in Fig. 2.14. Assuming noise with unknown statistics and false alarm probability upper bounded at 5×10^{-4}, the targets are classified with the overall probability of correct classification 0.9116.

Next, the face of a subject with varying illumination conditions has been shown to lie near a 9-dimensional subspace [35]. Thus a set of subjects can be assumed to lie near a union of subspaces. Using this assumption, for the Yale Database B [36], we first obtain subspace bases for each subject by using SVD on 18 randomly selected subject images. With these bases and assuming unknown noise statistics, we correctly identify subjects with probability 0.76 while upper bounding the false alarm rate at 1×10^{-3}.

The third dataset in consideration is the Hopkins 155 motion segmentation dataset [37], which consists of sequences of two and three motions extracted from several videos. It has been argued that different motion sequences extracted from tracking a set of points in a video lie in 3-dimensional subspaces [37]. We again use SVD on randomly selected sequences to learn the subspace bases. Using the UoS model with unknown noise statistics, the probability of correct classification over all sequences comes out to be 0.7664 by upper bounding the false alarm rate at 5×10^{-2}.

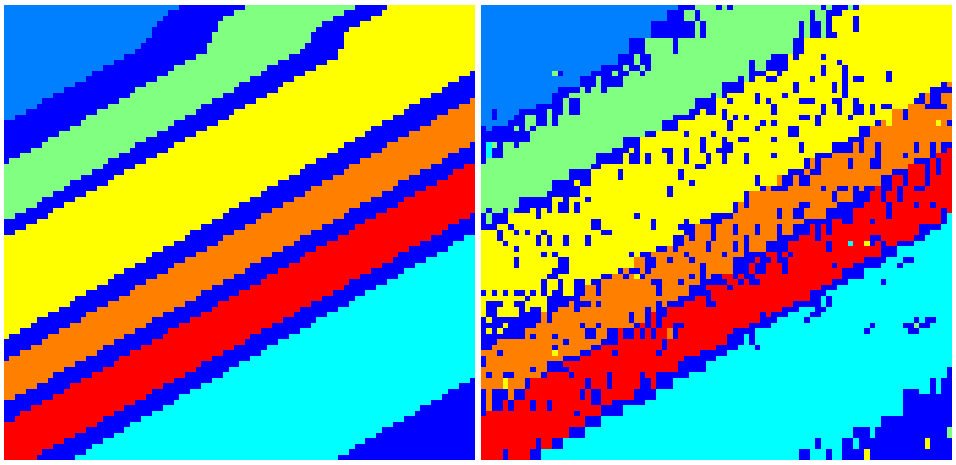

Figure 2.14: This figure shows the ground truth (left) for different classes in Salinas A scene and the detected targets (right) using the UoS detector under unknown noise statistics. The targets were detected with the classification accuracy of 91.16% when upper bounding the false alarm rate at 5×10^{-4}.

Next, recall that one of the main theses of this chapter is that the geometry of subspaces underlying a union impact the performance of active subspace detection. We now validate this claim on real-world data using the Salinas 'A' hyperspectral and the Hopkins motion

datasets. In the case of Salinas 'A' data, we select three targets whose underlying subspaces, when compared to other targets in the data, have increasing minimum principal angle and an increasing sum of principal angles (relative to the other subspaces). In the case of the Hopkins motion dataset, we select 11 sequences from the data in a similar fashion. We then carry out active subspace detection using the GLRTs derived in this chapter and report the results in Fig. 2.15 for the selected targets and sequences under the same SNR and detection thresholds. It can be seen from the figure that, even though the detection of the selected targets/sequences is carried out under identical conditions, the probability of correct classification of different targets/sequences varies as a function of the geometry of subspaces in the union. In particular, targets/sequences whose cumulative principal angles (relative to the subspaces of other targets/sequences) are larger result in higher probabilities of correct classification and vice versa. These results, coupled with the ones reported for synthetic data, confirm that geometry of subspaces play an integral role in the problem of active subspace detection under the UoS model.

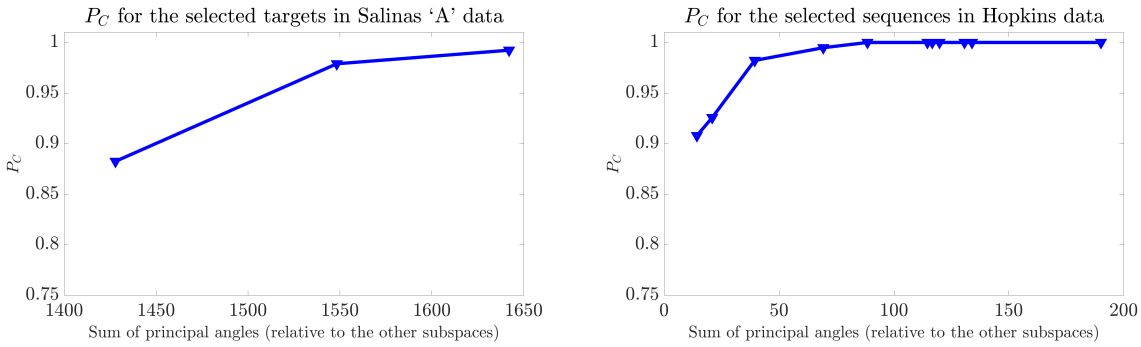

Figure 2.15: This figure shows the effects of geometry between subspaces for the Salinas 'A' hyperspectral and the Hopkins motion datasets. Three targets from Salinas 'A' data and 11 sequences from Hopkins motion data are selected such that they have increasing minimum and increasing cumulative principal angles with respect to the subspaces of other selected targets/sequences. One can see from the plots that target/subspaces (indicated with markers) having larger (cumulative) principal angles result in higher probabilities of correct classification (and vice versa).

2.5.3 Discussion

The experiments performed in Sec. 2.5.1 suggest that even though the bounds we obtain for probabilities of detection and correct classification are loose, they still predict the effect of subspace geometry on these probabilities correctly. In particular, we correctly predict that as the angles between whitened subspaces increase, the probabilities of detection and correct classification get higher, and vice-versa.

The results obtained in Sec. 2.5.2 for real-world datasets are not as good as some state-of-the-art algorithms (e.g., see [37]). However, there are certain advantages that our approach enjoys over the state-of-the-art methods. The first advantage is that our detection and classification methods allow control over the false alarm rate, which is not an option for other methods. Secondly, our method can work with just enough data, i.e., we just need enough samples to get good estimates of subspace bases and noise statistics. The third advantage is that our results explicitly cater to different levels of knowledge about the noise statistics and include that information in the detection and classification processes.

2.6 Conclusion

We introduced GLRTs for signal and active subspace detection under the UoS model. We analyzed the performance of the derived test statistics under various levels of knowledge about noise and explained the effect of colored noise geometry and geometry between subspaces on the detection and classification capabilities of these statistics. This was achieved by obtaining bounds on detection and classification probabilities in terms of the angles between subspaces and the angles that subspaces make with the noise eigenvectors. We also validated the insights of our analysis through Monte-Carlo experiments and experiments with real-world datasets.

2.7 Appendix

2.7.1 Proof of Theorem 1

In the case of the signal detection problem, the likelihoods under the two hypotheses are given by:

$$l_0(\mathbf{y}) \propto \exp\Big(-\frac{\mathbf{y}^T\mathbf{R}^{-1}\mathbf{y}}{2\sigma^2}\Big), \text{ and}$$
$$l_1(\mathbf{y}) \propto \exp\Big(-\frac{(\mathbf{y}-\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{x})}{2\sigma^2}\Big). \tag{2.18}$$

Since $\mathbf{x}$ is unknown in (2.18), we replace it with its *maximum likelihood* (ML) estimate $\widehat{\mathbf{x}}$, which is given by $\arg\min_k(\mathbf{y}-\mathbf{H}_k\boldsymbol{\theta})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}_k\boldsymbol{\theta})$, where $\mathbf{P}_{S_k} = \mathbf{H}_k(\mathbf{H}_k^T\mathbf{R}^{-1}\mathbf{H}_k)^{-1}\mathbf{H}_k^T\mathbf{R}^{-1}$ [27]. Consequently, the GLRT for this problem leads to the decision rule

$$\frac{l_1(\mathbf{y})}{l_0(\mathbf{y})} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma \;\Leftrightarrow\; \frac{\mathbf{y}^T\mathbf{R}^{-1}\mathbf{P}_{S_{\widehat{k}}}\mathbf{y}}{2\sigma^2} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma}, \tag{2.19}$$

where $\widehat{k} = \arg\max_k(\mathbf{y}^T\mathbf{R}^{-1}\mathbf{P}_{S_k}\mathbf{y})$, and $\bar{\gamma} = \log\gamma$ is the threshold used to control the probability of false alarm. Now, with appropriate substitutions, we can rewrite the final decision rule as: $T_{\mathbf{z}}^{2\sigma^2}\left(\mathbf{P}_{\bar{S}_{\widehat{k}}}\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma}$ with $\widehat{k} = \arg\max_k\ \mathbf{z}^T\mathbf{P}_{\bar{S}_k}\mathbf{z}$.

Similarly, the likelihoods under different hypotheses for the active subspace detection problem are given by:

$$l_0(\mathbf{y}) \propto \exp\Big(-\frac{\mathbf{y}^T\mathbf{R}^{-1}\mathbf{y}}{2\sigma^2}\Big), \text{ and}$$
$$l_k(\mathbf{y}) \propto \exp\Big(-\frac{(\mathbf{y}-\mathbf{H}_k\boldsymbol{\theta}_k)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}_k\boldsymbol{\theta}_k)}{2\sigma^2}\Big), \tag{2.20}$$

where $k = 1,\ldots,K_0$. Replacing the unknown $\boldsymbol{\theta}_k$'s in (2.20) with their ML estimates $\widehat{\boldsymbol{\theta}}_k = (\mathbf{H}_k^T\mathbf{R}^{-1}\mathbf{H}_k)^{-1}\mathbf{H}_k^T\mathbf{R}^{-1}\mathbf{y}$ [27] and comparing the generalized likelihoods lead to the

rule

$$\frac{l_{\widehat{k}}(\mathbf{y})}{l_0(\mathbf{y})} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \gamma \;\Leftrightarrow\; \frac{\mathbf{y}^T\mathbf{R}^{-1}\mathbf{P}_{S_{\widehat{k}}}\mathbf{y}}{2\sigma^2} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \bar{\gamma}. \tag{2.21}$$

Making the same substitutions as before, the final decision rule becomes: $T_{\mathbf{z}}^{2\sigma^2}\left(\mathbf{P}_{\bar{S}_{\widehat{k}}}\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \bar{\gamma}$. ■

2.7.2 Proof of Theorem 2

The probability of false alarm in the case of signal detection is given by:

$$\begin{aligned}
P_{FA} &= P_{\mathcal{H}_0}\Big(\widehat{\mathcal{H}}_1\Big) = P_{\mathcal{H}_0}\Big(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_{\widehat{k}}}) > \bar{\gamma}\Big) \\
&\overset{(a)}{=} \Pr\Big(T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_{\widehat{k}}}) > \bar{\gamma}\Big) = \Pr\Big(\bigcup_{k=1}^{K_0}\Big\{T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\Big\}\Big) \\
&= \sum_{k=1}^{K_0}\Pr\Big(T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\Big) - \\
&\sum_{k<j}^{K_0}\Pr\Bigg(\Big\{T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\Big\}\bigcap\Big\{T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_j}) > \bar{\gamma}\Big\}\Bigg) + \\
&+ \ldots + (-1)^{K_0-1}\Pr\Big(\bigcap_{k=1}^{K_0}\Big\{T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\Big\}\Big),
\end{aligned} \tag{2.22}$$

where (a) follows because $\mathbf{y}|\mathcal{H}_0 = \mathbf{n}$. We cannot evaluate (2.22) explicitly since it contains tail probabilities of k-tuples $\bigcap_{j=1}^{k}\Big\{\frac{\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\mathbf{w}}{2\sigma^2} > \bar{\gamma}\Big\}$, $k = 1, \ldots, K_0$. In particular, notice that $\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\mathbf{w}$ is a quadratic form of the variable $\mathbf{P}_{\bar{S}_j}\mathbf{w}$ and has a centered chi-squared distribution. This means that the distribution of the k-tuple is the joint distribution of k dependent chi-squared variables. These distributions exist in the literature for either independent quadratic forms or dependent quadratic forms under particular settings [38, 39, 40, 41]. However, the quadratic forms in (2.22) are neither independent nor fall under these

settings. We instead resort to upper bounding (2.22) by the union bound, i.e.,

$$
\begin{aligned}
P_{FA} &= \Pr\Big(\bigcup_{k=1}^{K_0}\Big\{T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\Big\}\Big) \\
&\leq \min\Big\{1\ ,\ \sum_{k=1}^{K_0}\Pr\Big(T_{\mathbf{w}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\Big)\Big\}. \qquad (2.23)
\end{aligned}
$$

Finally since, the null hypotheses for both signal and active subspace detection problems are the same, they end up having the same probability of false alarm.

Next, for the probability of detection P_D, note that

$$
\begin{aligned}
P_{S_k}\Big(\widehat{\mathcal{H}}_1\Big) &= P_{S_k}\Big(\bigcup_{i=1}^{K_0}\Big\{T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}\Big\}\Big) \\
&\overset{(b)}{=} \sum_{i=1}^{K_0} P_{S_k}\Big(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}\Big) \\
&- \sum_{i<j}^{K_0} P_{S_k}\Big(\Big\{T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}\Big\}, \Big\{T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_j}) > \bar{\gamma}\Big\}\Big) \\
&- \cdots + (-1)^{K_0-1} P_{S_k}\Big(\bigcap_{i=1}^{K_0}\Big\{T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}\Big\}\Big) \\
&\overset{(c)}{\leq} \min\Big\{1\ ,\ \sum_{i=1}^{K_0} P_{S_k}\Big(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_i}) > \bar{\gamma}\Big)\Big\}, \qquad (2.24)
\end{aligned}
$$

where (c) is again obtained using the union bound since the k-tuples in (b) cannot be expressed in closed form. Further, the lower bound in (2.6) follows from [42, Theorem 1].

Finally for the probability of classification P_C, we have:

$$
\begin{aligned}
P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) = P_{S_k}\Big(&\{T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}\}, \\
&\bigcap_{j=1, j\neq k}^{K_0} \{T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1\}\Big). \qquad (2.25)
\end{aligned}
$$

Since (2.25) cannot be evaluated explicitly as it involves *dependent definite and indefinite*

quadratic forms, we lower bound it by using the Fréchet inequalities [43]:

$$P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) \geq \max\Big\{0\ ,\ P_{S_k}(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}) + \sum_{j=1, j\neq k}^{K_0} P_{S_k}(T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1) - (K_0 - 1)\Big\}. \tag{2.26}$$

We conclude by noting that one could use [26, Lemma 1] to further lower bound (2.26). Specifically,

$$\begin{aligned} P_{S_k}(T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1) &= P_{S_k}(\mathbf{z}^T\mathbf{P}_{\bar{S}_j}^{\perp}\mathbf{z} - \mathbf{z}^T\mathbf{P}_{\bar{S}_k}^{\perp}\mathbf{z} > 0) \\ &= 1 - P_{S_k}(\mathbf{z}^T\mathbf{P}_{\bar{S}_j}^{\perp}\mathbf{z} - \mathbf{z}^T\mathbf{P}_{\bar{S}_k}^{\perp}\mathbf{z} < 0) \\ &\geq 1 - Q\big(\frac{1}{2}(1-2\eta_0)\sqrt{\lambda_{j\backslash k}}\big) - \Psi(n, \lambda_{j\backslash k}), \end{aligned} \tag{2.27}$$

where $\lambda_{j\backslash k} = \frac{1}{\sigma^2}\mathbf{z}^T\mathbf{P}_{\bar{S}_j}^{\perp}\mathbf{z}$ when $\mathbf{z} \in \bar{S}_k$. This leads to $P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k) \geq \max\big\{0, P_{S_k}(T_{\mathbf{z}}^{2\sigma^2}(\mathbf{P}_{\bar{S}_k}) > \bar{\gamma}) - \sum_{j:j\neq k} Q\big(\frac{1}{2}(1-2\eta_0)\sqrt{\lambda_{j\backslash k}}\big) - \sum_{j:j\neq k}\Psi(\eta_0, \lambda_{j\backslash k})\big\}$. ■

2.7.3 Proof of Theorem 3

The results derived in this appendix closely follow the derivations in [30]. The likelihood of $\boldsymbol{\xi}_p$ is given by:

$$l(\boldsymbol{\xi}_p) = \frac{1}{\sqrt{(2\pi)^m|\mathbf{R}|}}\exp\Big\{\frac{-1}{2}{\boldsymbol{\xi}_p}^T\mathbf{R}^{-1}\boldsymbol{\xi}_p\Big\}, \tag{2.28}$$

which is used to get the joint likelihoods under each hypothesis $\mathcal{H}_1$ and $\mathcal{H}_0$: $l_0(\mathbf{y}, \boldsymbol{\Xi})$ and $l_1(\mathbf{y}, \boldsymbol{\Xi})$, where $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_{N_0}]$. From these joint likelihoods, the ML estimate of $\mathbf{R}$ under $\mathcal{H}_1$ and $\mathcal{H}_0$ can be computed as $\widehat{\mathbf{R}}_1 = \frac{N_0}{N_0+1}\boldsymbol{\Sigma} + \frac{(\mathbf{y}-\mathbf{x})(\mathbf{y}-\mathbf{x})^T}{\sigma^2(N_0+1)}$ and $\widehat{\mathbf{R}}_0 = \widehat{\mathbf{R}}_1|_{\mathbf{x}=0}$, respectively.

Now, following the same steps as in the proof of Theorem 1, we can proceed to calculate the final decision rule for signal detection as $\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma}$, where $\widehat{k} = \arg\max_k(\widehat{\mathbf{z}}^T\widehat{\mathbf{P}}_{\bar{S}_k}\widehat{\mathbf{z}})$

and $\bar{\gamma} = \log \gamma$.

Next, note that the likelihood in (2.28) combined with the likelihoods in (2.20) also provide the joint likelihoods under each hypothesis for the active subspace detection problem. With trivial algebraic manipulations, the ML estimates of $\mathbf{R}$ in this case can be expressed as:

$$\begin{aligned}
\mathcal{H}_0 : \widehat{\mathbf{R}}_0 &= \frac{N_0}{N_0+1}\mathbf{\Sigma} + \frac{\mathbf{y}\mathbf{y}^T}{\sigma^2(N_0+1)}, \text{ and} \\
\mathcal{H}_k : \widehat{\mathbf{R}}_k &= \frac{N_0}{N_0+1}\mathbf{\Sigma} + \frac{(\mathbf{y}-\mathbf{x})(\mathbf{y}-\mathbf{x})^T}{\sigma^2(N_0+1)},
\end{aligned} \tag{2.29}$$

where $\mathbf{x}|\mathcal{H}_k = \mathbf{H}_k\boldsymbol{\theta}_k$. Using the ML estimates of $\mathbf{R}$ and the joint likelihoods, we can calculate the decision rule (similar to the proof of Theorem 1) as $\overline{T}_{\widehat{\mathbf{z}}}^{N_0\sigma^2}(\widehat{\mathbf{P}}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \bar{\gamma}$. ■

2.7.4 Proof of Theorem 5

This proof uses derivations from the proof of Theorem 3. The only additional estimate we need is for the variance σ^2 which can be found from the joint likelihoods with the estimate $\widehat{\mathbf{R}}$ substituted in them. This results in:

$$\begin{aligned}
\widehat{\sigma^2}|\mathcal{H}_1 &= \frac{N_0-m+1}{N_0 m}(\mathbf{y}-\mathbf{x})^T\mathbf{\Sigma}^{-1}(\mathbf{y}-\mathbf{x}), \text{and} \\
\widehat{\sigma^2}|\mathcal{H}_0 &= \frac{N_0-m+1}{N_0 m}\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y}.
\end{aligned} \tag{2.30}$$

The ML estimate of $\mathbf{x}$ in this case is the same as in the proof of Theorem 1. Putting these estimates together, we arrive at the final decision rule $T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \bar{\gamma}$, where $\widehat{k} = \arg\max_k(\widehat{\mathbf{z}}^T\widehat{\mathbf{P}}_{\bar{S}_k}\widehat{\mathbf{z}})$ and $\bar{\gamma} = \log \gamma$.

Similarly, the active subspace detection problem takes the same from as in Theorem 3 with an additional unknown variable σ^2. However, we can use the previously calculated ML estimates of σ^2, $\mathbf{R}$, and $\mathbf{x}$ to arrive at the final decision rule of $T_{\widehat{\mathbf{z}}}(\widehat{\mathbf{P}}_{\bar{S}_{\widehat{k}}}) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_{\widehat{k}}}{\gtrless}} \bar{\gamma}$. ■

2.7.5 Proof of Theorem 7

To get a better understanding of the parameters that effect the probability of correct classification, we analyze the terms $P_{S_k}(T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1)$ in (2.7) since these terms characterize the interactions between the whitened subspaces. Assuming $\mathbf{x} \in S_k$, notice that:

$$\begin{aligned}
&T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1 \Leftrightarrow \mathbf{z}^T\mathbf{P}_{\bar{S}_k}\mathbf{z} > \mathbf{z}^T\mathbf{P}_{\bar{S}_j}\mathbf{z} \\
&\Leftrightarrow (\bar{\mathbf{x}} + \mathbf{w})^T\mathbf{P}_{\bar{S}_k}(\bar{\mathbf{x}} + \mathbf{w}) > (\bar{\mathbf{x}} + \mathbf{w})^T\mathbf{P}_{\bar{S}_j}(\bar{\mathbf{x}} + \mathbf{w}) \\
&\overset{(a)}{\Leftrightarrow} \mathbf{w}^T\mathbf{P}_{\bar{S}_k}\mathbf{w} - \mathbf{w}^T\mathbf{P}_{\bar{S}_j}\mathbf{w} > \\
&\qquad - \bar{\mathbf{x}}^T\bar{\mathbf{x}} + \bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}} - 2\mathbf{w}^T\bar{\mathbf{x}} + 2\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}},
\end{aligned} \tag{2.31}$$

where $\bar{\mathbf{x}} = \mathbf{R}^{-\frac{1}{2}}\mathbf{x}$ is the whitened signal. We now focus on the quadratic forms $\bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}}$ and $\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}}$ in (2.31) because these are the terms where different subspaces interact with each other and that can be expressed in terms of the principal angles between whitened subspaces. Using the derivation provided in Appendix 2.7.6, we can bound $P_{S_k}(T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1)$ as:

$$\begin{aligned}
&P_{S_k}(\mathbf{w}^T\mathbf{P}_{\bar{S}_k}\mathbf{w} - \mathbf{w}^T\mathbf{P}_{\bar{S}_j}\mathbf{w} > -\bar{\mathbf{x}}^T\bar{\mathbf{x}} + \bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}} \\
&\qquad\qquad - 2\mathbf{w}^T\bar{\mathbf{x}} + 2\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}}) \\
&\geq P_{S_k}\Big(\|\mathbf{n}\|_2^2(\cos^2\psi_k - \cos^2\psi_j) > \\
&- \sum_{i=1}^{n}\theta_{ki}^2\sin^2\varphi_i^{(k,j)} + 2\sum_{i<p}^{n}|\theta_{ki}\theta_{kp}|\cos\varphi_i^{(k,j)}\cos\varphi_p^{(k,j)} \\
&+ \|\mathbf{n}\|_2\cos\psi_j\Big(\sum_{i=1}^{n}\theta_{ki}^2\cos^2\varphi_i^{(k,j)}\Big)^{\frac{1}{2}} - \|\mathbf{n}\|_2\cos\psi_k\Big(\sum_{i=1}^{n}\theta_{ki}^2\Big)^{\frac{1}{2}} \\
&+ \|\mathbf{n}\|_2\cos\psi_j\Big(2\sum_{i<p}^{n}|\theta_{ki}\theta_{kp}|\cos\varphi_i^{(k,j)}\cos\varphi_p^{(k,j)}\Big)^{\frac{1}{2}}\Big),
\end{aligned} \tag{2.32}$$

where $\varphi_i^{(k,j)}$ is the angle that $\mathbf{g}_i^k$ (i-th basis vector of whitened subspace $\bar{S}_k$, i.e., i-th column of $\mathbf{G}_k$) makes with the whitened subspace $\bar{S}_j$, $\varphi_{ip}^{(k,j)}$ is the angle between $\mathbf{g}_i^{k\to j}$ and

$\mathbf{g}_p^{k\to j}$ (i.e., the angles between the i-th and p-th basis vectors of whitened subspace $\bar{S}_k$ after projection onto the whitened subspace $\bar{S}_j$) and ψ_j is the angle between $\mathbf{w}$ and the whitened subspace $\bar{S}_j$.

This lower bound on $P_{S_k}(T_{\mathbf{z}}(\mathbf{P}_{\bar{S}_k}, \mathbf{P}_{\bar{S}_j}) > 1)$ is dependent on the principal angles $\varphi_i^{(k,j)}$ between the whitened subspace $\bar{S}_k$ and $\bar{S}_j$. In particular, we can see that as the principal angles $\varphi_i^{(k,j)}$ increase, the bound on the right hand side of the inequality (a) in (2.31) becomes smaller. This implies that lower bound on the tail probability in (2.32) becomes larger as the principal angles increase. This trend holds for all pairs of whitened subspaces $\bar{S}_j$ and $\bar{S}_k$ (for $j, k = 1, \ldots, K_0$ and $j \neq k$). This means that the lower bound for $P_{\mathcal{H}_k}(\widehat{\mathcal{H}}_k)$ in (2.7) also increases with increasing principal angles between the whitened subspaces.

We conclude by noting that this trend can also be derived from the lower bound expression in Remark 1. The quantities $Q()$ and $\Psi()$ in that expression are functions of $\lambda_{j\backslash i}$ and decrease monotonically as $\lambda_{j\backslash i}$ is increased [26]. This means that an increase in $\lambda_{j\backslash i}$ will result in an increase in the probability of correct classification. Since $\lambda_{j\backslash i}$ can be expressed as $\lambda_{j\backslash i} = \frac{1}{\sigma^2}\mathbf{z}^T\mathbf{P}_{\bar{S}_j}^{\perp}\mathbf{z} = \frac{1}{\sigma^2}\left(\mathbf{z}^T\mathbf{z} - \mathbf{z}^T\mathbf{P}_{\bar{S}_j}\mathbf{z}\right) = \frac{1}{\sigma^2}\left(\mathbf{z}^T\mathbf{z} - \bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}} - 2\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}} - \mathbf{w}^T\mathbf{P}_{\bar{S}_j}\mathbf{w}\right)$, one can use results from Appendix 2.7.6 to once again argue that as the angles between whitened subspaces increase, the lower bound on $\lambda_{j\backslash i}$ increases which in turn results in larger (lower) bound on the probability of correct classification. ■

2.7.6 Probability bound on ratio of quadratic forms

The outline of our procedure for deriving a lower bound on the probability of the comparison of quadratic forms is as follows: we first express $\bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}}$ and $\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}}$ in terms of the principal angles between whitened subspaces. We then obtain upper bounds on these quadratic forms that depend on the principal angles. Next we put these upper bounds in the expression for the probability of correct classification of the individual subspaces and finally we derive a lower bound on the probability of correct classification that is dependent on the principal angles between the whitened subspaces.

Let's consider $\bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}}$ when $\mathbf{x} \in S_k$:

$$\begin{aligned}
&\bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}} = \|\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}}\|_2^2 \overset{(a)}{=} \|\mathbf{P}_{\bar{S}_j}\mathbf{G}_k\boldsymbol{\theta}_k\|_2^2 \\
&\overset{(b)}{=} \|\theta_{k1}\mathbf{g}_1^{k\to j} + \cdots + \theta_{kn}\mathbf{g}_n^{k\to j}\|_2^2 \\
&\overset{(c)}{=} \sum_{i=1}^{n}\|\theta_{ki}\mathbf{g}_i^{k\to j}\|_2^2 + 2\sum_{i<p}^{n}\langle\theta_{ki}\mathbf{g}_i^{k\to j}, \theta_{kp}\mathbf{g}_p^{k\to j}\rangle, \\
&= \sum_{i=1}^{n}\theta_{ki}^2\|\mathbf{g}_i^k\|_2^2\cos^2\varphi_i^{(k,j)} + \\
&2\sum_{i<p}^{n}|\theta_{ki}|\|\mathbf{g}_i^k\|_2\cos\varphi_i^{(k,j)}|\theta_{kp}|\|\mathbf{g}_p^k\|_2\cos\varphi_p^{(k,j)}\cos\varphi_{ip}^{(k,j)}
\end{aligned} \tag{2.33}$$

where $\varphi_i^{(k,j)}$ are as defined in Appendix 2.7.5. Note that (a) in (2.33) follows from $\bar{\mathbf{x}} = \mathbf{G}_k\boldsymbol{\theta}_k$, (b) uses the notation $\mathbf{g}_i^{k\to j} = \mathbf{P}_{\bar{S}_j}\mathbf{g}_i$ and (c) uses the identity $\|\mathbf{a}+\mathbf{b}\|_2^2 = \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 2\langle\mathbf{a},\mathbf{b}\rangle$.

Now, if we assume $\mathbf{g}_i^k$'s to be the unit-norm principal vectors of $\bar{S}_k$, we can bound (2.33) as $\bar{\mathbf{x}}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}} \leq \sum_{i=1}^{n}\theta_{ki}^2\cos^2\varphi_i^{(k,j)} + 2\sum_{i<p}^{n}|\theta_{ki}|\cos\varphi_i^{(k,j)}|\theta_{kp}|\cos\varphi_p^{(k,j)}$. Similarly we have $\mathbf{w}^T\mathbf{P}_{\bar{S}_j}\bar{\mathbf{x}} \leq \|\mathbf{n}\|_2\cos\psi_j\Big(\sum_{i=1}^{n}\theta_{ki}^2\cos^2\varphi_i^{(k,j)}\Big)^{\frac{1}{2}} + \|\mathbf{n}\|_2\cos\psi_j\Big(2\sum_{i<p}^{n}|\theta_{ki}|\cos\varphi_i^{(k,j)}|\theta_{kp}|\cos\varphi_p^{(k,j)}\Big)^{\frac{1}{2}}$, where we have used the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ and ψ_j is the angle between $\mathbf{w}$ and the whitened subspace $\bar{S}_j$. Substituting these upper bounds in (2.31) we get:

$$\begin{aligned}
&\|\mathbf{n}\|_2^2(\cos^2\psi_k - \cos^2\psi_j) > -\sum_{i=1}^{n}\theta_{ki}^2\sin^2\varphi_i^{(k,j)} + \\
&2\sum_{i<p}^{n}|\theta_{ki}\theta_{kp}|\cos\varphi_i^{(k,j)}\cos\varphi_p^{(k,j)} + \\
&\|\mathbf{n}\|_2\cos\psi_j\Big(\sum_{i=1}^{n}\theta_{ki}^2\cos^2\varphi_i^{(k,j)}\Big)^{\frac{1}{2}} - \|\mathbf{n}\|_2\cos\psi_k\Big(\sum_{i=1}^{n}\theta_{ki}^2\Big)^{\frac{1}{2}} \\
&+ \|\mathbf{n}\|_2\cos\psi_j\Big(2\sum_{i<p}^{n}|\theta_{ki}\theta_{kp}|\cos\varphi_i^{(k,j)}\cos\varphi_p^{(k,j)}\Big)^{\frac{1}{2}},
\end{aligned} \tag{2.34}$$

which can be used to obtain (2.32). ■

CHAPTER 3
LEARNING PRODUCT GRAPHS UNDERLYING SMOOTH GRAPH SIGNALS

3.1 Introduction

Graph signal processing (GSP) is an emerging field in data science and machine learning that aims to generalize existing information processing methods to data that live on an irregular domain. This underlying irregular domain can be represented as a graph and analysis of signals on the vertices of this graph, aptly named graph signals, is enabled by the graph shift operator (GSO). Recent developments in GSP have already established that GSO-based data processing outperforms classical signal processing for several common tasks such as noise removal, signal filtering, wavelet representations, etc. [4, 5, 6, 7, 8]. The GSO is at the core of graph signal processing and could refer to either the adjacenecy matrix or one of the many types of Laplacian matrices associated with a graph. The exact choice of the GSO depends on the signal domain and the application of interest. The eigenvectors of GSO provide bases for the spectral analysis of graph signals and generalize the concepts of bandlimited signals to the graph domain [4, 5, 6, 7, 8]. GSO also facilitates the synthesis of graph-based filters [44, 45] and plays a pivotal role in the description of the notion of *smoothness* for graph signals [4, 5, 6, 7, 8].

The underlying graph (and hence the GSO) for some real-world datasets is either known apriori, or can trivially be constructed through domain knowledge. As an example, consider weather data collected over a region. In this example, different weather stations would act as nodes, their observations as graph signals, and one (possible) way to construct the graph would be to connect physically adjacent nodes. For most real-world data, however, such a trivially constructed graph is either non-optimal or it cannot be constructed in the first place due to lack of precise knowledge about the data generation process. This presents the

need to learn the true underlying graph from the data itself. In this regard, the problem of graph learning from the observed data (i.e., graph signals) has gained a lot of attention in the recent years [44, 46, 45, 47, 15, 48, 49, 50, 4].

Graph learning refers to the problem of learning an unknown underlying graph from observed graph signals by exploiting some property of the graph signals. Traditional approaches for graph learning have proposed algorithms whose best-case complexity scales quadratically with the number of nodes in the graph [49, 15, 51, 47, 48]. These approaches might be suitable for learning small graphs, but even for moderately sized graphs the learning cost would be prohibitive. Moreover, for learning an arbitrary graph (Laplacian), the number of parameters one needs to learn also scale quadratically with the number of nodes. Both of these problems hinder the amenability of traditional graph learning approaches to large-scale real-world datasets. Our work on graph learning, in contrast, hinges on the fact that real-world data is often generated over graphs that have an additional inherent structure. This inherent structure is dictated by either the way the data is acquired, by the arrangement of the sensors, or by the inherent relation of variables being observed [6]. Moreover, this inherent structure of the graph being considered also presents itself in the associated GSO, which can incidentally be represented in terms of the product of several smaller *factor* GSOs. In this chapter, we will focus on three such structures that can be described in terms of three different products termed Kronecker, Cartesiasn and strong products. Although aware of the presence of these product structures in real-world graphs (and the associated GSOs) [6], the research community has yet to propose algorithms that incorporate the graph product structure in the graph learning procedure. Additionally, as the number of free parameters scales quadratically with the number of nodes in the graph, given the massive nature of the datasets available today, it has become imperative to devise methods that fully utilize the product structure of graphs to reduce the number of parameters to be learned. Moreover, posing the problems in terms of smaller factor graphs instead of the graph itself can enable efficient data representation [6], and result in reduced sample

complexity as one has to learn fewer parameters. To this end, the main objective of this work is to investigate the problem of learning product graphs from data in an efficient and scalable manner.

3.1.1 Prior work

The existing works in graph signal processing can mainly be divided into four chronological categories. The first set of works in GSP introduced the idea of information processing over graphs [7, 5, 8]. These works highlight the advantages and superior performance of graph-based signal processing approach (with known underlying graph) over classical signal processing. The second wave of research in this area built upon the first one to exploit knowledge of the underlying graph for graph signal recovery from samples obtained over a subset of graph nodes or from noisy observations of all nodes [16, 52]. Through these works, the idea of bandlimitidness was extended to graph signals and the concept of smooth graph signals was introduced. Since the underlying graph is not always available beforehand, the third wave of GSP analyzed the problem of recovering the underlying graph through observations over the graph [44, 46, 45, 47, 15, 48, 49]. Finally, the fourth wave of research in GSP has focused on joint identification of the underlying graph and graph signals from samples/noisy observations using interrelated properties of graph signals and graphs [50, 53, 54, 49].

Within the third set of papers in GSP, our work falls in the category of combinatorial graph Laplacian estimation [47, 15, 48, 49] from graph signals. Combinatorial graph Laplacian refers to the unnormalized Laplacian of an unstructured graph with no self loops [48]. The earliest of works in this category [49] aims to jointly denoise noisy graph signals observed at the nodes of the graph and also learn the graph from these denoised graph signals. The authors pose this problem as a multiconvex problem in the graph Laplacian and the denoised graph signals, and then solve it via an alternating minimization approach that solves a convex problem in each unknown. The authors in [46] examine the problem of

graph Laplacian learning when the eigenvectors of the graph Laplacian are known beforehand. They achieve this by formulating a convex program to learn a valid graph Laplacian (from a feasible set) that is diagonalized by the noiseless and noisy Laplacian eigenvectors.

The work in [47] takes a slightly different route and learns a sparse unweighted graph Laplacian matrix from noisy graph signals through an alternating minimization approach that restricts the number of edges in the graph. In contrast to earlier work, [44] focuses on learning a graph diffusion process from observations of stationary signals on graphs through convex formulations. In this regard, the authors also consider different criteria in addition to searching for a valid Laplacian and devise specialized algorithms to infer the diffusion processes under these criteria. In [51, 15] the graph learning problem is adressed by posing it in terms of learning a sparse weighted adjacency matrix from the observed graph signals. Finally, the authors in [48] provide a comprehensive unifying framework for inferring several types of graph Laplacians from graph signals. They also make connection with the state-of-the-art and describe where the past works fit in light of their proposed framework.

It should be mentioned here that since Laplacian matrices (and thus adjacency matrices) are related to precision matrices, defined as the (pseudo-)inverses of covariance matrices, imposing a structure on the graph adjacency matrix amounts to imposing a structure on the covariance of data. Earlier works in the field have already made comparisons of Laplacian learning approaches with those for learning precision matrices from data [49, 48], and established the superior graph recovery performance of Laplacian-based learning. Some recent works have also investigated learning structured covariance and precision matrices, and their usefulness in efficiently representing real-world datasets [55, 56, 57]. While these models work well in practice, we will demonstrate through our experiments that there are scenarios where graph-based learning outperforms structured covariance-based learning (see Sec. 4.5 for details).

3.1.2 Our contributions

Our first contribution in this work is a novel formulation of the graph learning problem as a linear program. We show, both theoretically and empirically, that graph adjacency matrices can be learned through a simple and fast linear program. We then shift our attention towards learning structured graphs. Prior works regarding graph learning have only considered arbitrary graphs with either some connectivity constraints [15], or no constraints at all [45, 49]. In all cases, the complexity of the graph learning procedure and the number of free parameters scale quadratically with the number of nodes in the graph, which can be prohibitively large in real-world scenarios. In contrast, our work focuses on inferring the underlying graph from graph signals in the context of structured graphs. Specifically, we investigate graphs that can be represented as Kronecker, Cartesian, and strong products of several smaller graphs. We first show how, for these product graphs, the graph adjacency matrix, the graph Laplacian, the graph Fourier transform, and the graph smoothness measure can be represented with far fewer parameters than required for arbitrary graphs. This reduction in number of parameters to be learned results in reduced sample complexity and helps avoid overfitting. Afterwards, we outline an algorithm to learn these product graphs from the data and provide convergence guarantees for the proposed algorithm in terms of the estimation error of factor graphs. We validate the performance of our algorithm with numerical experiments on both synthetic and real data.

3.1.3 Organization

The rest of this chapter is organized as follows. In Sec. 3.2 we give a probabilistic formulation of the graph learning problem in line with existing literature. Then, we propose our novel formulation of the graph learning problem as a liner program in Sec. 3.3. Sec. 3.4 describes the motivation for product graphs and formulates the graph learning problem in the context of product graphs. In Sec. 3.5 we propose an algorithm for learning product graphs from data and derive error bounds on estimated factor graphs. We present our numerical

experiments with synthetic and real datasets in Sec. 4.5, and the chapter is concluded in Sec. 3.7.

3.2 Probabilistic problem formulation

In this section we formulate the arbitrary graph learning problem from a probabilistic standpoint. Let as assume access to $m = 1, \ldots, M_0$ graph signals $\mathbf{x}_m \in \mathbb{R}^n$ observed on n nodes of an undirected graph $G = \{V, E\}$ without any self loops, where V and E represent the nodes and edges of the graph. The weighted edges of this graph G can be represented as a weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, which has a zero diagonal owing to no self-loops in the graph. Based on the adjacency matrix $\mathbf{W}$, one can define the degree matrix $\mathbf{D} = \mathrm{diag}(W\mathbf{1})$, which is a diagonal matrix containing the weighted degree of each node at the respective diagonal entry. The associated unnormalized graph Laplacian for G can then be defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The adjacency matrix $\mathbf{W}$ of the graph can be decomposed as $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ and its eigenvectors define the graph Fourier basis for the graph Fourier transform [6].

The signals observed on the nodes of a graph are assumed to have a joint distribution given by a multivariate normal distribution, i.e., $\mathbf{x}_m \sim \mathcal{N}(0, \mathbf{L}^\dagger)$, where $\mathbf{L}^\dagger$ is the pseudoinverse of $\mathbf{L}$ and $\mathbf{L}$ represents the graph Laplacian. In words, signals generated over a graph can be seen as being generated over a Gaussian Markov Random Field (GMRF) whose precision matrix is the graph Laplacian [49]. Given independent observations $\{\mathbf{x}_m\}$, the maximum likelihood estimate (MLE) of $\mathbf{L}$ can be expressed as:

$$\begin{aligned}\widehat{\mathbf{L}} &= \arg\max_{\mathbf{L} \in \mathcal{L}} |\mathbf{L}|^{\frac{M_0}{2}} \exp(-\frac{1}{2}\sum_{m=1}^{M_0} \mathbf{x}_m^T \mathbf{L} \mathbf{x}_m) \\ &= \arg\min_{\mathbf{L} \in \mathcal{L}} -\log|\mathbf{L}| + \frac{1}{M_0}\sum_{m=1}^{M_0} \mathbf{x}_m^T \mathbf{L} \mathbf{x}_m, \end{aligned} \tag{3.1}$$

where $\mathcal{L}$ represents the class of valid Laplacians, i.e., a symmetric positive semi-definite

matrix with rows that sum to zero and nonpositive off-diagonal entries. With the Laplacian constraints, the problem in (3.1) can be further expressed as:

$$\begin{aligned}\widehat{\mathbf{L}} = \ & \arg\min_{\mathbf{L}} \ -\log|\mathbf{L}| + \frac{1}{M_0}\sum_{m=1}^{M_0}\mathbf{x}_m^T\mathbf{L}\mathbf{x}_m \\ & \text{s.t. } \mathbf{L}\mathbf{1} = \mathbf{0},\ \mathrm{trace}(\mathbf{L}) = n,\ (\mathbf{L})_{ij} = (\mathbf{L})_{ji} \leq 0. \end{aligned} \tag{3.2}$$

Interactions in the real world tend to be mostly local, and thus not all nodes in a graph are connected to each other in real-world datasets. To impose only local interactions, usually a sparsity term regularizing the off-diagonal entries of the Laplacian matrix is added to the graph learning objective to learn *sparse* graphs. Therefore, traditional graph learning approaches [51, 49, 56, 57] take a form similar to the following:

$$\begin{aligned}\widehat{\mathbf{L}} = \ & \arg\min_{\mathbf{L}} \ -\log|\mathbf{L}| + \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathbf{x}_m^T\mathbf{L}\mathbf{x}_m + \beta\|\mathbf{L}\|_{1,\text{off}} \\ & \text{s.t. } \mathbf{L}\mathbf{1} = \mathbf{0},\ \mathrm{trace}(\mathbf{L}) = n,\ (\mathbf{L})_{ij} = (\mathbf{L})_{ji} \leq 0, \end{aligned} \tag{3.3}$$

where $\|\mathbf{L}\|_{1,\text{off}}$ represents a sparsity penalty on the off-diagonal entries of $\mathbf{L}$, the parameter $\alpha > 0$ controls the penalty on the quadratic term, and the parameter $\beta > 0$ controls the density of the graph. In the following section, we show that the traditional graph learning problem can be significantly simplified and that arbitrary graphs can actually be learned through a simple linear program.

3.3 Graph learning as a linear program

Let us start by inspecting the traditional graph learning problem in (3.3). In particular, let us first focus on the term $\log|\mathbf{L}|$ in the objective function and the constraint $\mathrm{trace}(\mathbf{L}) = n$. We can express this log-determinant term in the objective as $\log|\mathbf{L}| = \sum_{i=1}^{n}\log\lambda_i$, where λ_i is the i-th largest eigenvalue of $\mathbf{L}$. Thus, through this $\log|\mathbf{L}|$ term, the MLE constrains the

spectrum of the Laplacian matrix to be estimated. However, for our problem of estimating the Laplacian matrix, the constraint $\mathrm{trace}(\mathbf{L}) = \sum_{i=1}^{n} \lambda_i = n$, is already putting a hard constraint on the sum of eigenvalues of the Laplacian matrix. Moreover, the constraint $\mathbf{L1} = \mathbf{0}$ is forcing the smallest eigenvalue of the Laplacian to be zero. In the presence of these constraints, the log-determinant regularization in the objective function is no longer necessary to arrive at a valid Laplacian matrix or to avoid trivial solutions. Another advantage of removing the log-determinant term is the massive savings in computational complexity as this term forces one to employ singular value decomposition at each step of the learning algorithm [56, 57].

Let us also examine the term $\sum_{m=1}^{M_0} \mathbf{x}_m^T \mathbf{L} \mathbf{x}_m$ in the objective in (3.1). This term comes from the likelihood of the observed signals with the Laplacian as the precision matrix, and also represents the sum of Dirichlet energy or "smoothness" of the observed graph signals [51, 49, 48]. It has been shown in the existing literature [51] that this term can be expressed as a weighted sparsity regularization on the graph adjacency matrix as $\sum_{m=1}^{M_0} \mathbf{x}_m^T \mathbf{L} \mathbf{x}_m = \mathrm{trace}(\mathbf{X}^T \mathbf{L} \mathbf{X}) = \|\mathbf{W} \circ \mathbf{Z}\|_1$. Here $\mathbf{X}$ is the data matrix with $\mathbf{x}_m$ as the m-th column, and $\mathbf{Z}$ is the matrix of pairwise distances between rows of $\mathbf{X}$ such that (i, j)-th entry in $\mathbf{Z}$ is the euclidean distance between the i-th and j-th row of $\mathbf{X}$. This implies that the sum of Dirichlet energy in the objective implicitly regularizes the sparsity of $\mathbf{W}$ and thus controls the density of edges in the graph. Therefore, presence of this term in the objective eliminates the need to explicitly regularize the sparsity of the graph to be learned.

In light of the preceding discussion, we propose to solve the following linear program [58] for learning graphs:

$$\begin{aligned} \widehat{\mathbf{L}} = \min_{\mathbf{L}} \quad & \frac{\alpha}{M_0} \sum_{m=1}^{M_0} \mathbf{x}_m^T \mathbf{L} \mathbf{x}_m \\ \text{s.t. } & \mathbf{L1} = \mathbf{0}, \ \mathrm{trace}(\mathbf{L}) = n, \ (\mathbf{L})_{ij} = (\mathbf{L})_{ji} \leq 0, \end{aligned} \tag{3.4}$$

where α is a regularization parameter that controls the smoothness of the graph signals and

thus the sparsity of edges in the graph.

3.3.1 Fast solver for the graph learning linear program

We now present an algorithm, named **G**raph learning with **L**inear **P**rogramming (GLP), for solving the linear graph learning problem (3.4). To proceed, note that the objective term in the graph learning problem can be reformulated as:

$$\begin{aligned}
\frac{1}{M_0}\sum_{m=1}^{M_0}\mathbf{x}_m^T\mathbf{L}\mathbf{x}_m &= \frac{1}{M_0}\sum_{m=1}^{M_0}(\mathbf{x}_m^T\mathbf{D}\mathbf{x}_m - \mathbf{x}_m^T\mathbf{W}\mathbf{x}_m) \\
&= \frac{1}{M_0}\sum_{m=1}^{M_0}(\mathbf{x}_m^T\mathrm{diag}(\mathbf{W}\mathbf{1})\mathbf{x}_m - \mathbf{x}_m^T\mathbf{W}\mathbf{x}_m) \\
&= \frac{1}{M_0}\sum_{m=1}^{M_0}(\mathbf{x}_m^T\mathrm{diag}(\mathbf{x}_m)\mathbf{W}\mathbf{1} - \mathbf{x}_m^T\mathbf{W}\mathbf{x}_m) \\
&= \frac{1}{M_0}\sum_{m=1}^{M_0}(\bar{\mathbf{x}}_m^T\mathbf{W}\mathbf{1} - \mathbf{x}_m^T\mathbf{W}\mathbf{x}_m) \\
&= \frac{1}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}(\bar{\mathbf{x}}_m^T\mathbf{W}\mathbf{1} - \mathbf{x}_m^T\mathbf{W}\mathbf{x}_m) \\
&= \frac{1}{M_0}\mathrm{trace}\Big[\mathbf{W}\sum_{m=1}^{M_0}\mathbf{1}\bar{\mathbf{x}}_m^T - \mathbf{W}\sum_{m=1}^{M_0}\mathbf{x}_m\mathbf{x}_m^T\Big] \\
&= \mathrm{trace}\Big[\mathbf{W}\Big(\sum_{m=1}^{M_0}\mathbf{1}\bar{\mathbf{x}}_m^T - \sum_{m=1}^{M_0}\mathbf{x}_m\mathbf{x}_m^T\Big)/M_0\Big] \\
&= \mathrm{trace}(\mathbf{W}\widetilde{\mathbf{S}}) = \mathrm{vec}(\widetilde{\mathbf{S}})^T\mathrm{vec}(\mathbf{W}) = \widetilde{\mathbf{s}}^T\mathbf{M}\mathbf{w}, \qquad (3.5)
\end{aligned}$$

where the matrix $\widetilde{\mathbf{S}} = (\sum_{m=1}^{M_0}\mathbf{1}\bar{\mathbf{x}}_m^T - \sum_{m=1}^{M_0}\mathbf{x}_m\mathbf{x}_m^T)/M_0$, the vector $\bar{\mathbf{x}}_m^T = \mathbf{x}_m^T\mathrm{diag}(\mathbf{x}_m) = \mathbf{x}_m^T \circ \mathbf{x}_m^T$, $\mathrm{vec}(\mathbf{W}) = \mathbf{M}\mathbf{w}$, $\mathbf{w}$ is a vector of distinct elements from upper triangular part of the symmetric matrix $\mathbf{W}$, and $\mathbf{M}$ is the duplication matrix that duplicates the elements from $\mathbf{w}$ to generate a vectorized version of $\mathbf{W}$. With this rearrangement of the objective and the

adjacency matrix, our graph learning problem can be posed as follows:

$$\begin{aligned} \widehat{\mathbf{w}} = \ & \min_{\mathbf{w}} \alpha\, \widetilde{\mathbf{s}}^T \mathbf{M}\mathbf{w} \\ & \text{s.t. } \mathbf{A}\mathbf{w} = \mathbf{b},\ \mathbf{w}_i \geq 0, i \in F, \end{aligned} \tag{3.6}$$

where $\mathbf{A}$ is a matrix that represents the equality constraints from (3.4) in terms of equality constraints on $\mathbf{w}$, $\mathbf{b} = [\mathbf{0}^T, n]^T$, and F is the set containing the indices of the off-diagonal elements in $\mathbf{w}$. Once the solution $\widehat{\mathbf{w}}$ is obtained, it can be converted to the symmetric adjacency matrix $\widehat{\mathbf{W}}$, which can then be used to get $\widehat{\mathbf{L}}$.

The standard way of solving a linear program with mixed (equality and inequality) constraints is through interior point methods whose complexity scales quadratically with the problem dimension [58]. A better alternative is to deploy a first-order method whose per-iteration complexity is linear in the number of nonzero entries of $\mathbf{A}$. However, a first-order method would exhibit slow convergence for a linear program because of the lack of smoothness and strong convexity in linear programs [59]. To overcome these issues, linear programs have been solved through the Alternating Direction Method of Multiplier (ADMM) [60, 59]. To solve our proposed linear formulation of graph learning, we follow a recent algorithm proposed in [59]. This ADMM-based algorithm for linear programs proposed a new variable splitting scheme that achieves a convergence rate of $\mathcal{O}(\|\mathbf{A}\|^2 \log(1/\epsilon))$, where ϵ is the desired accuracy of the solution. To this end, we start by modifying the original graph learning problem with the introduction of an additional variable $\mathbf{y}$ as follows:

$$\begin{aligned} \widehat{\mathbf{w}} = \ & \min_{\mathbf{w}} \mathbf{c}^T \mathbf{w} \\ & \text{s.t. } \mathbf{A}\mathbf{w} = \mathbf{b},\ \mathbf{w} = \mathbf{y},\ \mathbf{y}_i \geq 0, i \in F, \end{aligned} \tag{3.7}$$

where $\mathbf{c} = \alpha \mathbf{M}^T \widetilde{\mathbf{s}}$. The corresponding augmented Lagrangian can then be expressed as

Algorithm 1: : GLP—ADMM for graph learning with linear programming

Input: Observations $\{\mathbf{x}_m\}_{m=1}^{M_0}$, maximum iterations T_0, and parameter $\alpha, \rho > 0$
Initialize: $\mathbf{y}^{(1)} \leftarrow \mathbf{0}$, $\mathbf{z}^{(1)} \leftarrow \mathbf{1}$
  for $t = 1$ to T_0
    $\mathbf{e}^{(t+1)} \leftarrow -\mathbf{A}_{\mathbf{w}}^T[\mathbf{z}^{(t)} + \rho(\mathbf{A}_{\mathbf{y}}\mathbf{y}^{(t)} - \widetilde{\mathbf{b}})] - \mathbf{c}$
    $\mathbf{w}^{(t+1)} \leftarrow \rho^{-1}(\mathbf{I} + \mathbf{A}^T\mathbf{A})^{-1}\mathbf{e}^{(t+1)}$
    $\mathbf{y}^{(t+1)} \leftarrow [\mathbf{w}^{(t+1)} + \mathbf{z}_{\mathbf{y}}^{(t)}/\rho]_{\underset{F}{\geqslant}\mathbf{0}}$
    $\mathbf{z}^{(t+1)} \leftarrow \mathbf{z}^{(t)} + \rho(\mathbf{A}_{\mathbf{w}}\mathbf{w}^{(t+1)} + \mathbf{A}_{\mathbf{y}}\mathbf{y}^{(t+1)} - \widetilde{\mathbf{b}})$
  end
Output: Final adjacency estimate $\widehat{\mathbf{w}} \leftarrow \mathbf{w}^{(t+1)}$.

follows:

$$L(\mathbf{w}, \mathbf{y}, \mathbf{z}) = \mathbf{c}^T\mathbf{w} + h(\mathbf{y}) + \mathbf{z}^T(\mathbf{A}_{\mathbf{w}}\mathbf{w} + \mathbf{A}_{\mathbf{y}}\mathbf{y} - \widetilde{\mathbf{b}}) + \rho/2\|\mathbf{A}_{\mathbf{w}}\mathbf{w} + \mathbf{A}_{\mathbf{y}}\mathbf{y} - \widetilde{\mathbf{b}}\|_2^2, \quad (3.8)$$

where $h(\mathbf{y})$ denotes the non-negativity constraint on the entries of $\mathbf{y}$ indexed by F, i.e., $\forall i \in F$, $h(\mathbf{y}) = 0$ when $y_i \geq 0$, and $h(\mathbf{y}) = \infty$ when $y_i < 0$. Moreover, $\mathbf{z} = [\mathbf{z}_{\mathbf{w}}^T, \mathbf{z}_{\mathbf{y}}^T]^T$, $\mathbf{z}_{\mathbf{w}}$ and $\mathbf{z}_{\mathbf{y}}$ are the Lagrange multipliers, $\mathbf{A}_{\mathbf{w}} = [\mathbf{A}^T, \mathbf{I}]^T$, $\mathbf{A}_{\mathbf{y}} = [\mathbf{0}, -\mathbf{I}]^T$, and finally $\widetilde{\mathbf{b}} = [\mathbf{b}^T, \mathbf{0}^T]^T$. One can then use ADMM to go through the steps outlined in Algorithm 1 until convergence to obtain $\widehat{\mathbf{w}}$.

In Algorithm 1, $[\cdot]_{\underset{F}{\geqslant}\mathbf{0}}$ is entrywise thresholding that projects the entries with indices in F to the nonnegative orthant. As we can see from the algorithm, all updates have closed-form solutions and the most computationally expensive step is the $\mathbf{w}^{(t+1)}$ update that involves matrix inversion. This matrix inversion, however, can be computed efficiently using the identity $(\mathbf{I} + \mathbf{A}^T\mathbf{A})^{-1} = \mathbf{I} - \mathbf{A}^T(\mathbf{I} + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}$. Since matrix $\mathbf{A}$ is a fat matrix, $\mathbf{A}\mathbf{A}^T$ has smaller dimensions than $\mathbf{A}^T\mathbf{A}$. Moreover, one can easily see that $\mathbf{A}\mathbf{A}^T$ is a matrix of

dimensions $(n+1) \times (n+1)$, and

$$\mathbf{A}\mathbf{A}^T = \begin{bmatrix} c_n & 1 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix}. \tag{3.9}$$

where $c_n = 2n^2 - n$. In addition, the inverse only needs to be computed once at the start of the algorithm since this matrix is deterministic and depends only on the size of the adjacency matrix being estimated.

3.3.2 Parameter and computational complexities

The number of parameters that one needs to learn a graph adjacency matrix is $\frac{n(n+1)}{2}$. This implies that the number of unknown parameters scales quadratically with the number of nodes in the graph. Additionally, the per-iteration computational complexity of the proposed method also scales quadratically with the number of nodes [59]. The same computational and memory complexities also hold for the existing state-of-the-art graph learning algorithms [49, 15, 51, 47, 48]. However, while these complexities are manageable for small graphs, for real-world datasets with even hundreds of nodes the current methods become prohibitive. To overcome these issues, we will next examine the problem of learning product graphs from data.

3.4 Why product graphs?

In this section we briefly review product graphs and their implications towards graph learning. We investigate how product graphs provide a way to efficiently represent graphs with a huge number of nodes, and we revisit the notion of smoothness of signals over product graphs.

Let us consider K_0 graphs $G_k = \{V_k, E_k\}$, for $k = 1, \ldots, K_0$, where V_k and E_k represent the vertices and edges of the k-th graph. The product of these graphs would result in a product graph $G = \{V, E\}$, with V and E representing the vertices and edges of the resultant graph. The three most commonly investigated graph products and their respective adjacency matrices are discussed below. Note that graph adjacency matrices are considered in this work because each kind of product structure is directly reflected in adjacency matrix of the resultant graph.

3.4.1 Kronecker graphs

For the Kronecker product of graphs G_k, for $k = 1, \ldots, K_0$, with adjacency matrices $\mathbf{W}_k$, the Kronecker product graph can be expressed as $G = \bigotimes_{[K_0]} G_k = G_{K_0} \otimes G_{K_0-1} \otimes \cdots \otimes G_1$. The respective Kronecker-structured adjaceny matrix of the resultant graph can written in terms of component/factor adjacency matices as $\mathbf{W} = \bigotimes_{[K_0]} \mathbf{W}_k$. Additionally, if the factor adjacency matrix $\mathbf{W}_k$ can be expressed via eigenvalue decomposition (EVD) as $\mathbf{W}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^T$, then the Kronecker adjaceny matrix can be written as (using the properties of the Kronecker product [6]):

$$\begin{aligned}\mathbf{W} &= (\mathbf{U}_{K_0} \mathbf{\Lambda}_{K_0} \mathbf{U}_{K_0}^T) \otimes \cdots \otimes (\mathbf{U}_1 \mathbf{\Lambda}_1 \mathbf{U}_1^T) \\ &= (\bigotimes_{[K_0]} \mathbf{U}_k)\,(\bigotimes_{[K_0]} \mathbf{\Lambda}_k)\,(\bigotimes_{[K_0]} \mathbf{W}_k^T) = \mathbf{U} \mathbf{\Lambda}_{\text{kron}} \mathbf{U}^T. \end{aligned} \tag{3.10}$$

One can see that both the eigenmatrix and the eigenvalue matrix of the Kronecker adjacency matrix have a Kronecker structure in terms of the component eigenmatrices and component eigenvalue matrices, respectively. Given the number of edges in the component graphs are $|E_k|$, the number of edges in the Kronecker graph are $|E| = 2^{K_0-1} \prod_{k=1}^{K_0} |E_i|$.

An example of Kronecker product graph is the bipartite graph of a recommendation system like Netflix [61] where the graph between users and movies can be seen as a Kronecker product of two smaller factor graphs. In fact, the adjacency matrix of any bipartite graph

can be represented in terms of a Kronecker product of appropriate factor matrices [62]. As adjacency matrices are also closely related to precision matrices (i.e. inverse covariance matrices), and inverse of Kronecker product is Kronecker product of inverses [6], imposing Kronecker structure on the adjacency matrix also amounts to imposing a Kronecker structure on the covariance matrix of the data.

The optimization problem in (3.4) can be specialized to the case of learning Kronecker graphs by explicitly imposing the Kronecker product structure on the adjacency matrix and posing the problems in terms of the individual factor adjacency matrices, rather than the bigger adjacency matrix produced after the product. This leads us to the following nonconvex problem for learning Kronecker graphs:

$$\min_{\{\mathbf{W}_k \in \mathcal{W}\}_{k=1}^{K_0}} \frac{\alpha}{M_0} \text{trace}\left[\left[\bigotimes_{[K_0]} \mathbf{W}_k \right] \left(\sum_{m=1}^{M_0} \mathbf{1}\bar{\mathbf{x}}_m^T - \sum_{m=1}^{M_0} \mathbf{x}_m \mathbf{x}_m^T \right) \right]. \tag{3.11}$$

3.4.2 Cartesian graphs

The Cartesian product (also called Kronecker sum product) of graphs G_k is represented as $G = \bigoplus_{[K_0]} G_k = G_{K_0} \oplus G_{K_0-1} \oplus \cdots \oplus G_1$. The correspoding cartesian adjacency matrix can be written in terms of the component adjacency matrices as $\mathbf{W} = \bigoplus_{[K_0]} \mathbf{W}_k$. Furthermore, with the EVD of the component adjacency matrices, the Cartesian adjacency can be decomposed as [6]:

$$\begin{aligned} \mathbf{W} &= (\mathbf{U}_{K_0} \mathbf{\Lambda}_{K_0} \mathbf{U}_{K_0}^T) \oplus \cdots \oplus (\mathbf{U}_1 \mathbf{\Lambda}_1 \mathbf{U}_1^T) \\ &= \left(\bigotimes_{[K_0]} \mathbf{U}_k \right) \left(\bigoplus_{[K_0]} \mathbf{\Lambda}_i \right) \left(\bigotimes_{[K_0]} \mathbf{W}_k^T \right) = \mathbf{U} \mathbf{\Lambda}_{\text{cart}} \mathbf{U}^T. \end{aligned} \tag{3.12}$$

This means that the eigenmatrix and the eigenvalue matrix of the Cartesian adjacency matrix are represented, respectively, as Kronecker and Cartesian products of component eigenmatrices and eigenvalue matrices. The number of edges in the Cartesian graph can be found as $|E| = \sum_{k=1}^{K_0} (\bigotimes_{[K_0]\setminus k} n_i) |E_k|$, where $|E_k|$ represents the number of edges and n_k represents

the number of vertices in the k-th component graph.

A typical exmaple of a Cartesian product graph is images. Images reside on two dimensional rectangular grids that can be represented as the Cartesian product between two line graphs pertaining to the rows and columns of the image [6]. A social network can also be approximated as a Cartesian product of an inter-community graph with an intra-community graph [6].

Similar to the previous discussion, the optimization problem in (3.4) can be specialized to learning Cartesian graphs by explicitly imposing the Cartesian structure and posing the problem in terms of the factor adjacency matrices as follows:

$$\min_{\{\mathbf{W}_k \in \mathcal{W}\}_{k=1}^{K_0}} \frac{\alpha}{M_0} \text{trace}\left[\left[\bigoplus_{[K_0]} \mathbf{W}_k \right] \left(\sum_{m=1}^{M_0} \mathbf{1}\bar{\mathbf{x}}_m^T - \sum_{m=1}^{M_0} \mathbf{x}_m \mathbf{x}_m^T \right) \right]. \tag{3.13}$$

3.4.3 Strong graphs

The strong product of graphs G_k can be represented as $G = \boxtimes_{[K_0]} G_k = G_{K_0} \boxtimes G_{K_0-1} \boxtimes \cdots \boxtimes G_1$. The respective strong adjacency matrix of the resultant strong graph is given in terms of the component adjacency matrices as $\mathbf{W} = \boxtimes_{[K_0]} \mathbf{W}_k$, and can be further expressed as:

$$\begin{aligned} \mathbf{W} &= (\mathbf{U}_{K_0} \mathbf{\Lambda}_{K_0} \mathbf{U}_{K_0}^T) \boxtimes \cdots \boxtimes (\mathbf{U}_1 \mathbf{\Lambda}_1 \mathbf{U}_1^T) \\ &= \left(\bigotimes_{[K_0]} \mathbf{U}_k \right) \left(\boxtimes_{[K_0]} \mathbf{\Lambda}_i \right) \left(\bigotimes_{[K_0]} \mathbf{W}_k^T \right) = \mathbf{U} \mathbf{\Lambda}_{\text{str}} \mathbf{U}^T \end{aligned} \tag{3.14}$$

in terms of EVD of the component adjacency matrices.

The strong product graphs can be seen as the sum of Kronecker and Cartesian products of the factor adjacency matrices. An example of data conforming to the strong product graph is a spatiotemporal sensor network graph, which consists of a strong product of a spatial graph and a temporal graph (representing the temporal dependencies of the sensors). The spatial graph has as many nodes as the number of sensors in the sensor network and

represents the spatial distribution of sensors. On the other hand, the temporal graph has as many nodes as the number of temporal observations of the whole sensor network and represents the overall temporal dynamics (changes in connectivity over time) of the network [6].

By making the strong product structure explicit in terms of the factor adjacency matrices, the optimization problem for learning strong graphs can be expressed as the following nonconvex problem:

$$\min_{\{\mathbf{W}_k \in \mathcal{W}\}_{k=1}^{K_0}} \frac{\alpha}{M_0} \text{trace}\left[\left[\boxtimes_{[K_0]} \mathbf{W}_k \right] \left(\sum_{m=1}^{M_0} \mathbf{1}\bar{\mathbf{x}}_m^T - \sum_{m=1}^{M_0} \mathbf{x}_m \mathbf{x}_m^T \right) \right]. \tag{3.15}$$

3.4.4 Product graph Fourier transform

One can see from (3.10),(3.12), and (3.14) that the graph Fourier transform of a product graph (which is the eigenmatrix of the product adjacency matrix), has a Kronecker structure in terms of the eigenmatrices of the component graph adjacency matrices: $\mathbf{U} = \bigotimes_{[K_0]} \mathbf{U}_k$. In terms of the implementation of the graph Fourier transform, this structure provides an efficient implementation of the graph Fourier as (using the properties of Kronecker product and tensors [18]):

$$\mathbf{U}^T \mathbf{x} = (\bigotimes_{[K_0]} \mathbf{U}_k)^T \mathbf{x} = \text{vec}(\mathcal{X} \times_{[K_0]} \mathbf{U}_k), \tag{3.16}$$

where $\mathbf{x} \in \mathbb{R}^{n_1 n_2 \ldots n_{K_0}}$ is an arbitrary graph signal on the product graph, and $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_{K_0}}$ represents appropriately tensorized version of the signal $\mathbf{x}$. Because of this, one does not need to form the huge Fourier matrix $\mathbf{U}$ and can avoid costly matrix multiplications by just applying the component graph Fourier matrices to each respective mode of the tensorized observation $\mathcal{X}$ and then vectorizing the result.

3.4.5 Smoothness

Smoothness of a graph signal is one of the core concepts in graph signal processing [4, 5, 6, 7, 8] and product graph Laplacians provide an efficient representation for the notion of smoothness. The smoothness of a graph signal can be measured through the Dirchlet energy defined as $\mathbf{x}^T\mathbf{L}\mathbf{x}$. The Dirichlet energy can be reexpressed as: $\mathbf{x}^T\mathbf{L}\mathbf{x} = \mathbf{x}^T(\mathbf{D} - \mathbf{W})\mathbf{x} = \mathbf{x}^T\mathbf{D}\mathbf{x} - \mathbf{x}^T\mathbf{W}\mathbf{x}$. Let us now focus on each term separately in the context of product graphs. For the term involving $\mathbf{W}$ we have:

$$\begin{aligned}\mathbf{x}^T\mathbf{W}\mathbf{x} &= \mathbf{x}^T\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{x} = (\mathbf{U}^T\mathbf{x})^T\mathbf{\Lambda}(\mathbf{U}^T\mathbf{x}) \\ &= \mathrm{vec}(\mathcal{X} \underset{[K_0]}{\times} \mathbf{U}_k)^T\mathbf{\Lambda}\mathrm{vec}(\mathcal{X} \underset{[K_0]}{\times} \mathbf{U}_k). \end{aligned} \tag{3.17}$$

Similarly, the term involving $\mathbf{D}$ can be reexpressed as:

$$\begin{aligned}\mathbf{x}^T\mathbf{D}\mathbf{x} &= \mathbf{x}^T\mathrm{diag}(\mathbf{W}\mathbf{1})\mathbf{x} = \mathbf{x}^T\mathrm{diag}(\mathbf{x})\mathbf{W}\mathbf{1} \\ &= (\mathbf{x} \odot \mathbf{x})^T\mathbf{W}\mathbf{1} = \bar{\mathbf{x}}^T\mathbf{W}\mathbf{1}, \end{aligned} \tag{3.18}$$

which can be computed efficiently along the lines of (3.17). With this reformulation, one circumvents the need to explicitly form the prohibitively large eigenmatrix $\mathbf{U}$ and can evaluate the Dirichlet energy much more efficiently with just mode-wise products with the smaller component eigenmatrices.

3.4.6 Representation complexity

Let us consider an unknown graph G with the number of nodes $|V| = n = \prod_{k=1}^{K_0} n_k$, where n_k represents the number of nodes in each component graph and K_0 is total number of component graphs. If one were to learn this graph by means of an arbitrary adjacency matrix, the number of parameters that need to estimated would be $\frac{n(n+1)}{2}$ (since the graph adjacency matrix is a symmetric matrix). On the other hand, for the same graph, by uti-

lizing the product model of the graph adjacency matrix, one would need to estimate only $\sum_{k=1}^{K_0} \frac{n_k(n_k+1)}{2}$ parameters. This means that, e.g., $n_1 = n_2 = \cdots = n_{K_0} = \bar{n}$, imposing the product structure on graph adjacency matrix reduces the number of parameters needed to be learned by $\bar{n}^{K_0-1}/K_0$.

3.5 Algorithm for learning product graphs

In the previous section we highlighted some properties and advantages of product graphs and we posed the optimization problems for learning these graphs. We now propose an algorithm for solving these product graph learning problems. To this end, we first recognize that even though these problems posed are nonconvex (except for Cartesian graphs), the factor-wise minimization problems for any factor adjacency matrix is still convex if all the other factors are fixed. Moreover, these factor-wise can be solved through Algorithm 1 proposed in the earlier sections. These observations lead us to propose a block coordinate descent (BCD) based algorithm, named BPGL (BCD for product graph learning), that minimizes over each factor adjacency matrix in cyclic fashion. The proposed algorithm is provided in Algorithm 2, and in the following discussion we present the factor-wise problems for each product graph.

Algorithm 2: : BPGL–BCD for product graph learning

Input: Observations $\{\mathbf{x}_m\}_{m=1}^{M_0}$, maximum iterations N_0, and parameter α
Initialize: $\{\widehat{\mathbf{W}}_k\}_{k=1}^{K_0}$
  for $n = 1$ to N_0
    for $k = 1$ to K_0
      while stopping criteria
        Solve (3.19), (3.20), or (3.21) for $\widehat{\mathbf{W}}_k$ via Algorithm 1
      end
    end
  end
Output: Final adjacency matrix estimates $\widehat{\mathbf{W}}_k$.

3.5.1 Kronecker graphs

Since Algorithm 2 utilizes factor-wise minimization, we can characterize the error for product graph learning in terms of the factor-wise error of each factor adjacency matrix (while keeping the other factors fixed). The factor-wise minimization problem in the case of learning Kronecker graphs boils down to (see Appendix 3.8.1):

$$\min_{\mathbf{W}_k \in \mathcal{W}} \alpha \operatorname{trace}(\mathbf{D}_k \bar{\mathbf{S}}_k) - \alpha \operatorname{trace}\big(\mathbf{W}_k \mathbf{S}_k\big) \tag{3.19}$$

for $k = 1, \cdots, K_0$, and where $\mathbf{S}_k$ and $\bar{\mathbf{S}}_k$ are as defined in Appendix 3.8.1. As pointed out before, each of these factor-wise problem is a convex program. The error characteristics of factor-wise problems are provided in the following theorem with the proof in Appendix 3.8.1.

Theorem 8. *For the k-th adjacency factor comprising a Kronecker product adjacency matrix, while keeping other components $\mathbf{W}_j$ for $j = 1, \cdots, K_0, j \neq k$ fixed, with high probability, the error between the sample-based minimization with M_0 samples and the population-based minimization of* (3.19) *satisfies $\mathcal{O}\Big(\frac{n_k^2 \log(n_k)}{nM_0}\Big)$, for an appropriate α. Moreover, also with high probability, the error between the estimated factor $\widehat{\mathbf{W}}_k$ and the true factor $\mathbf{W}_k$ satisfies $\|\widehat{\mathbf{W}}_k - \mathbf{W}_k\|_F = \mathcal{O}\left(\sqrt{\frac{n_k \log(n_k)}{nM_0}}\right)$.*

3.5.2 Cartesian graphs

For Cartesian graphs, the factor-wise minimization problems, for $k = 1, \cdots, K_0$, can be represented as follows (see Appendix 3.8.2):

$$\min_{\mathbf{W}_k \in \mathcal{W}} \alpha \operatorname{trace}(\mathbf{D}_k \bar{\mathbf{T}}_k) - \alpha \operatorname{trace}\big(\mathbf{W}_k \mathbf{T}_k\big), \tag{3.20}$$

where $\mathbf{T}_k$ and $\bar{\mathbf{T}}_k$ are as defined in Appendix 3.8.2. As before, each factor-wise problem is a convex program, and the following theorem characterizes the factor-wise minimization

of the graph learning problem for Cartesian graphs.

Theorem 9. *The objective function* (3.13) *for the Cartesian graph learning problem is convex, and can be represented as a sum of terms that are linear in each factor adjacency matrix. Moreover, each factor-wise minimization satisfies the same error bounds as from Theorem 8.*

The proof of this theorem is given in Appendix 3.8.2. The theorem states that the objective function for learning Cartesian product graphs is convex and separable in each factor adjacency matrix, i.e., the objective function can be represented as a sum of linear terms each of which is dependent on only one factor adjacency matrix. Therefore, for learning Cartesian graphs, this allows one to optimize over all factors in parallel, unlike the problems for learning other product graphs.

3.5.3 Strong graphs

The problem for learning strong graphs can be posed factor-wise, for $k = 1, \cdots, K_0$, (see Appendix 3.8.3) as follows:

$$\min_{\mathbf{W}_k \in \mathcal{W}} \alpha \operatorname{trace}(\mathbf{D}_k \bar{\mathbf{Z}}_k) - \alpha \operatorname{trace}\big(\mathbf{W}_k \mathbf{Z}_k\big), \tag{3.21}$$

where $\mathbf{Z}_k$ and $\bar{\mathbf{Z}}_k$ are as defined in Appendix 3.8.3. The following theorem, with its proof in Appendix 3.8.3, characterizes the behavior of factor-wise minimization problems for strong graphs.

Theorem 10. *For the k-th adjacency factor comprising a strong product adjacency matrix, while keeping other components $\mathbf{W}_j$ for $j = 1, \cdots, K_0, j \neq k$ fixed, the error between the sample-based minimization with M_0 samples and the population-based minimization of* (3.21) *satisfies $\mathcal{O}\Big(\frac{n_k^2 \log(n_k)}{nM_0}\Big)$ for an appropriately chosen α, with high probability. Moreover, with high probability, the error between the estimated factor $\widehat{\mathbf{W}}_k$ and the true factor $\mathbf{W}_k$ satisfies $\|\widehat{\mathbf{W}}_k - \mathbf{W}_k\|_F = \mathcal{O}\left(\sqrt{\frac{n_k \log(n_k)}{nM_0}}\right)$.*

Remark 5. Theorems 8,9 and 10 claim that the estimated factor lies within a ball of radius $\frac{\log(n_k)}{(n/n_k)M_0}$ around the true factor. The accuracy of the estimate increases with the number of available observations M_0, and the product of the dimensions of the other factors $n/n_k = \prod_{j \neq k} n_j$. Moreover, the accuracy decreases with the increasing dimensions of the factor being estimated.

Remark 6. The theorems 8,9 and 10 provide the estimation error bounds of factor adjacency matrices for respective product graphs. Error bounds on the product adjacency matrix are non-trivial and will be the focus of future work. However, intuitively speaking, the error for estimating Cartesian graphs should be smaller than other product graphs as the Cartesian adjacency matrix can be obtained as a linear combination of the factor matrices, whereas Kronecker and strong graphs contain terms obtained through products of the factor adjacency matrices (which compounds the error multiplicatively).

3.5.4 Convergence properties

Each of the preceding theorems in this chapter derive the error bounds after the first iteration of Algorithm 2 for each product structure. The overall convergence of the algorithm can be established through the following theorem:

Theorem 11. *The product graph learning algorithm Algorithm 2 is guaranteed to converge to a stationary point at a linear rate.*

The proof of this theorem is provided in Appendix 3.8.4.

3.5.5 Computational complexity

The computational complexity of solving each factor-wise problem scales quadratically with the number of nodes in the graph. This implies that when the product structure is imposed, one only has to solve K_0 smaller problems each with computational complexity of $\mathrm{O}(\bar{n}^2)$, assuming the special case of $n_1 = n_2 = \cdots = n_{K_0} = \bar{n}$. In contrast, for learning

unstructured graphs the computational complexity would scale as $\mathrm{O}(n^2) = \mathrm{O}(\prod_{k=1}^{K_0} n_k^2) = \mathrm{O}(\bar{n}^{2K_0})$. Thus, the computational gains are huge in comparison to the original problem for learning unstructured graphs!

3.5.6 Error bound for arbitrary graphs

As a byproduct of Theorem 8, we can also obtain an error bound for arbitrary graph learning problem. Following along the lines of Theorem 8, we can say that by solving (3.4) one is guaranteed to converge to the true adjacency matrix of the unstructured graph with the error given as: $\|\widehat{\mathbf{W}} - \mathbf{W}^*\|_F = \mathcal{O}_P\Big(\sqrt{\log(n)/M_0}\Big)$.

Remark 7. Taking a closer look at the error bounds for learning arbitrary and structured graphs reveals an important point. The denominator in the error bound is the number of observations available to estimate the graph. For M_0 observed graph signals, the number of observations available to arbitrary graph learning are (obviously) M_0; however, for estimating the k-th factor adjacency when learning product graphs the effective number of observations are $\prod_{j\neq k} n_j \times M_0$. This means that imposing the product structure results in an increased number of effective observations to estimate each factor adjacency matrix. This combined with the reduced number of parameters required to learn these graphs makes product graphs very attractive for real world applications.

3.6 Numerical experiments

This section provides results for learning product graphs from synthetic and real datasets. We first present experiments for learning arbitrary graphs through our proposed linear program in Sec. 3.3, and then the results for learning products graphs from synthetic data through Algorithm2. Afterwards we validate the performance of our proposed algorithm for product graphs on real-world datasets.

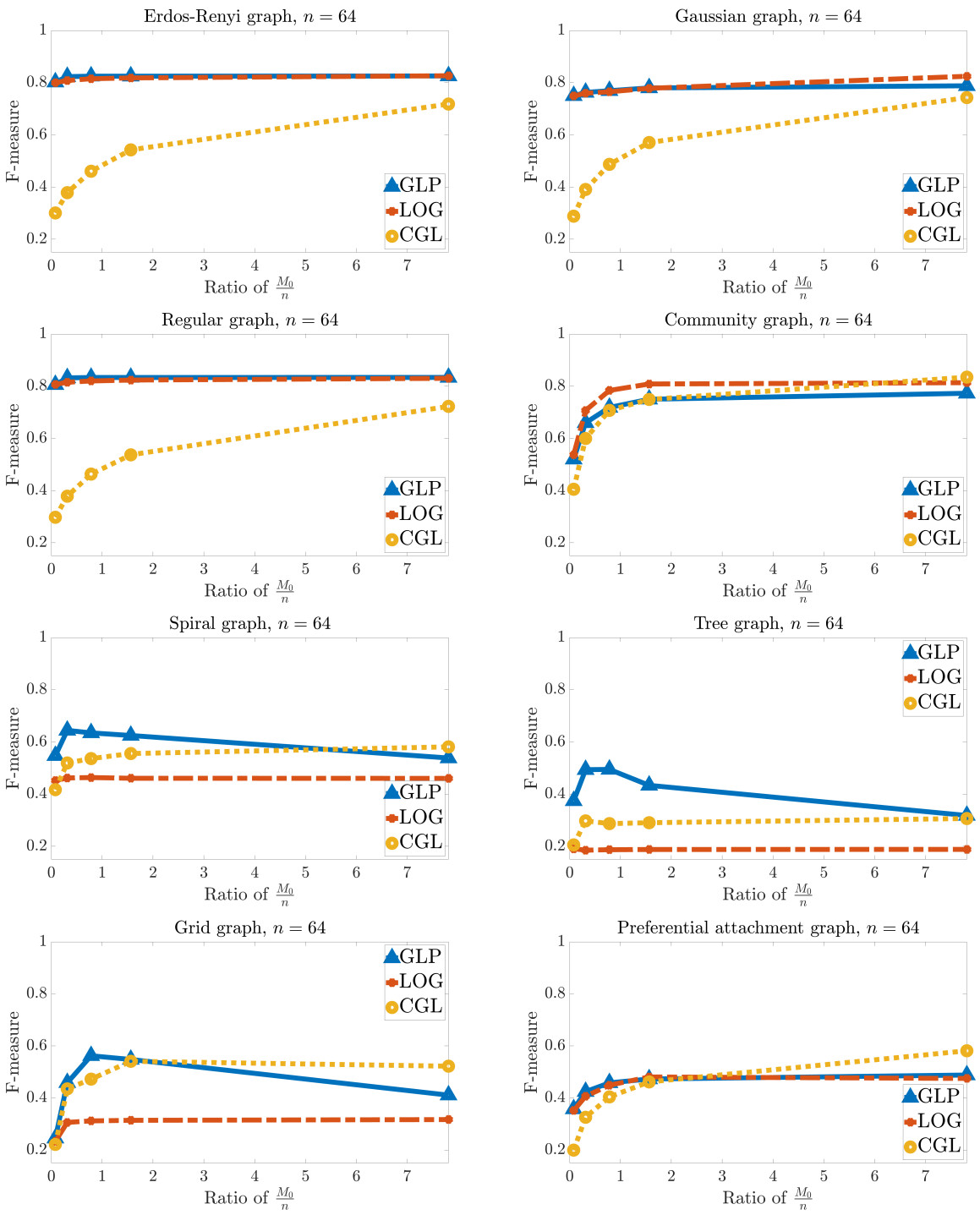

Figure 3.1: F-measure values for various graphs for our proposed graph learning algorithm (GLP), LOG [48], and CGL [15].

3.6.1 Synthetic data: Arbitrary graphs

To showcase the performance of our new formulation for graph learning, we run synthetic experiments on a graph with $n = 64$ nodes. We generate various different graph types such as: (i) a sparse random graph with Gaussain weights, (ii) an Erdos-Renyi random graph with edge probability 0.7, (iii) a scale-free graph with preferential attachment (6 edges at each step), (iv) a random regular graph where each node is connected to $0.7n$ other nodes, (v) a uniform grid graph, (vi) a spiral graph , (vii) a community graph , and (viii) a low stretch tree graph on a grid of points. The related details of how the graphs are simulated can be found in [63, 49]. For each kind of graph, we generate 20 different realizations, and for each realization we generate observations using a degenerate multivariate Gaussian distribution with the graph Laplacian as the precision matrix [49, 48, 51].

We compare the performance of our proposed method with two other state-of-the-art methods for arbitrary graph learning: (i) combinatorial graph learning from [48] (which we refer to as CGL), and (ii) graph learning method from [15] (which we refer to as LOG), which also aims to learn a combinatorial graph Laplacian through a slightly different optimization problem than [48]. We choose α for our algorithm in the range $0.75^i\sqrt{\frac{\log(n)}{M_0}}$ with the integer i in the range $[0, 14]$, as dictated by the error bounds for learning graphs in Appendix 3.8.1 and by the existing literature [49, 48, 51]. Furthermore, we choose $\rho = 0.75/\log(M_0)$ through empirical evaluation. For each algorithm, in the prescribed range of the optimization parameters, we manually choose the parameters that produce the best results.

The results of our experiments are shown as F-measure values in Fig. 3.1. F-measure is the harmonic mean of precision and recall, and signifies the overall accuracy of the algorithm [48]. Precision here denotes the fraction of true graph edges recovered among all the recovered edges, and recall signifies the fraction of edges recovered from the true graph edges. One can see that our algorithm (except for community graphs) performs just as well or better than the existing state-of-the-art algorithms. Moreover, the average

performance over all graphs in Fig. 3.2 shows that on average we outperform the existing algorithms. A runtime comparison of all algorithms in Fig. 3.2 also reveals competitive run time for our proposed scheme. The runtime of LOG is the smallest, however, this algorithm has a huge computational overhead for the first step which is done separately from the main algorithm. This overhead relates to the construction of a matrix of pairwise distances of all rows of data matrix $\mathbf{X}$. In contrast, other algorithms work with the graph signal observations directly and do not require extra steps.

Remark 8. For some graphs in Fig. 3.1, the performance for GLP seems to worsen as number of observations grow. We have seen empirically, that this is due to the limited range that we have considered for searching the optimization parameter. For a bigger range, this downward trend is likely to disappear as one can choose a more appropriate parameter over this range. The range that we have prescribed is the one mostly used in the literature and on average works well in most settings.

3.6.2 Synthetic data: Product graphs

We now present the results of our numerical experiments involving synthetic data for product graphs. We run experiments for random Erdos-Renyi factor graphs with $n = n_1 n_2 n_3 = 12 \times 12 \times 12$ nodes, and having either Cartesian, Kronecker or strong structure. We then use our proposed algorithms to learn the generated graphs with varying number of observations and compare the performance with the algorithm in [15] as its performance was the second best in Fig. 3.1. The results for all three types of product graphs are shown in Fig. 3.3 (top). For a fixed number of observations, Cartesian product graphs can be learned with the highest F-measure score followed by strong and then Kronecker graphs. The figure also shows that for each graph, imposing product structure on the learned graph drastically improves the performance of the learning algorithm. Fig. 3.3 (bottom) also shows the run times comparison of our approach BPGL with the algorithm in [15]. Even for a graph of this size, with total number of nodes $n = 1728$, we can see a considerable reduction in

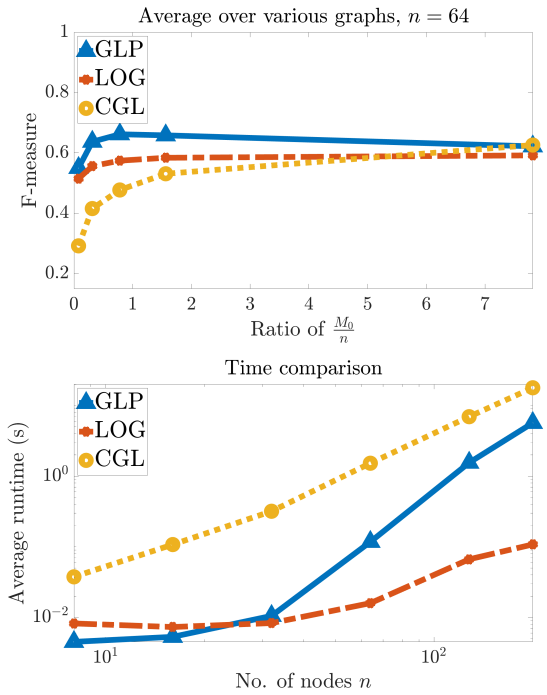

Figure 3.2: Average F-measure values over all graphs (left) from Fig. 3.1 for our proposed graph learning algorithm (GLP), LOG [48], and CGL [15]. Average run times over 30 trials for each algorithm (right), with increasing number of nodes.

run times. Thus, our learning algorithm that explicitly incorporates the product structure of the graph enjoys superior performance, reduced computational complexity and faster run times.

3.6.3 United States wind speed data

The first real data we use for experimentation is NCEP wind speed data. The NCEP wind speed data [64] represents wind conditions in the lower troposphere and contains daily averages of U (east-west) and V (north-south) wind components over the years 1948-2012. Similar to the experiments in [57] with preprocessed data, we use a grid of $n_1 n_2 = 10 \times 10 = 100$ stations to extract the data available for the United States region. From this extracted data, we choose the years 2003-2007 for training and keep the years 2008-2012 for testing purposes. Using a non-overlapping window of length $n_3 = 8$, which amounts to dividing the data into chunks of 8 days, we obtain $M_0 = 228$ samples, each of length $n =$

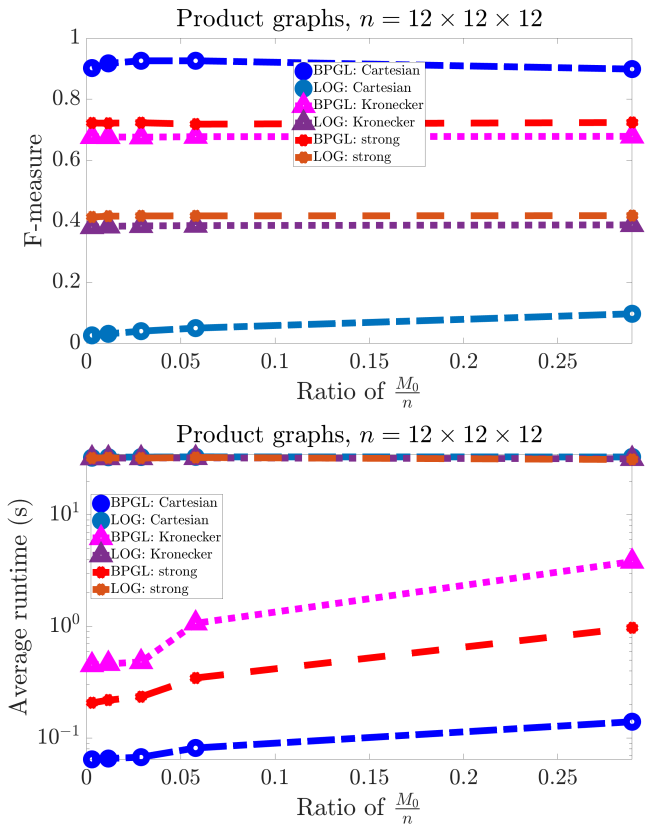

Figure 3.3: Precision, recall and F-measure values for various values of the β parameter. The plots shown are for Cartesian (top), Kronecker (middle), and strong (bottom) graphs when using only 5 observations for learning.

$n_1 n_2 n_3$. Therefore, for training purposes we have 228 samples, where each sample contains spatiotemporal data for 100 stations over 8 days. Same amount of data is obtained for testing through the same procedure. The testing procedure consists of introducing missing values in each sample of the test data by omitting the data for the 8th day, and then using a using a linear minimum-mean-square-error (LMMSE) estimator [65, Chapter 4] to predict the missing values. Our proposed method estimates the (structured) adjacency matrix of the graph (which is related to the precision matrix of the data), and we use $\mathbf{W} + \mathbf{I}$ in place of the data covariance for the LMMSE estimator.

We make a comparison of the following: (1) sample covariance matrix (SCM), (2) the permuted rank-penalized least squares (PRLS) approach [57] with $r = 6$ Kronecker components, (3) PRLS with $r = 2$ Kronecker components, (4) time varying graph learning

Table 3.1: Comparison of prediction RMSE for US wind speed data

Method	**RMSE reduction over SCM (dB)**	**parameters**
SCM	–	320400
TVGL [66]	1.0461	40656
PRLS [57] ($r = 6$)	1.7780	30492
PRLS [57] ($r = 2$)	-1.5473	10164
BPGL strong	**1.8640**	**5082**
BPGL Cartesian	1.3105	5082

Comparison of our graph learning method with SCM, PRLS and TVGL. Our proposed graph learning method outperforms the existing methods for learning sum of Kronecker structures covariance matrix from the data and for learning time varying graph learning. Moreover, our proposed procedure outperforms while using considerably fewer parameters.

(TVGL) approach from [66] which was shown to outperform the approach in [67], (5) spatiotemporal strong graph with BPGL with a spatial component of size $n_1 n_2$ and a temporal component of size n_3, and (6) spatiotemporal Cartesian graph with BPGL of the same dimensions. The parameters for PRLS were chosen for optimal performance as given in [57]. The optimization parameters for TVGL and BPGL were manually tuned for best performance. It should be noted here that SCM and PRLS aim to learn a covariance matrix and a structured covariance matrix from the data, respectively.

SCM aims to estimate $\frac{n(n+1)}{2}$ parameters, while the number of parameters that PRLS needs to estimate is $r(\frac{n_1 n_2(n_1 n_2+1)}{2} + \frac{n_3(n_3+1)}{2})$. On the other hand, TVGL aims to estimate $n_3(\frac{n_1 n_2(n_1 n_2+1)}{2})$ parameters, while BPGL needs to learn only $\frac{n_1 n_2(n_1 n_2+1)}{2} + \frac{n_3(n_3+1)}{2}$ parameters for both strong and Cartesian graphs. The mean prediction root-mean-squared errors (RMSE) for all methods are shown in Table 3.1. One can see that our proposed method outperforms PRLS and TVGL while estimating far fewer parameters than both. The table also shows that learning a strong graph for this data results in a higher RMSE reduction over the baseline (SCM), and is thus better suited for this data than the Cartesian product graph.

3.6.4 ABIDE fMRI data: Exploratory data analysis

The second real data that we use as an application for our proposed algorithm is a part of the ABIDE fMRI dataset [68, 69]. Our aim is to learn the graphs over the fMRI data of control and autistic subjects and to use the learned graphs to higlight the differences in the control and autistic brains. The data we obtain is already preprocessed to remove various fMRI artifacts and for controlization of the obtained scans [70]. The final preprocessed data consists of measurements from $n_1 = 111$ brain regions scanned over 116 time instances for each subject. The data contains scans for control and autistic subjects, and to avoid class imbalance we randomly choose 47 subjects for each class. Out of the 47 subjects for each class, we then randomly choose 30 subjects for training and keep the remaining 17 for testing purposes. We use a non-overlapping window length of $n_2 = 29$ which results into $M_0 = 120$ samples of length $n = n_1 \times n_2$.

As before, we compare the performance of our proposed approach with SCM and PRLS. Table 3.2 shows the results of our experiments. One can see that our approach performs very similar to PRLS for both Cartesian and strong product graphs, all the while using much fewer parameters (five times fewer). We also see that strong product graphs are more suited to model brain activity. The work in [70] suggests that autistic brains exhibit hypoconnectivity in different regions of the brain as compared to control subjects. The results from our graph learning procedure go a step further and bring more insight into the spatiotemporal dynamics of the brain. Firstly, as already suggested in [70], we see clear evidence of spatial hypoconnectivity (see Fig. 3.4). More importantly, our learned graphs in Fig. 3.5 reveal that, in addition to spatial hypoconnectivity, autistic brains also suffer from temporal hypoconnectivity.

3.6.5 Estrogen receptor data

The final dataset that we experiment on is the estrogen receptor data [71, 72], which consists of 157 samples of 693 probe sets related to estrogen receptor pathway. We aim to

Table 3.2: Comparison of prediction RMSE for ABIDE fMRI data

Method	**RMSE reduction over SCM (dB)**	**parameters**
SCM	–	5182590
PRLS Normal	**2.1793**	33255
Cartesian GL Control	2.0980	6651
Strong GL Control	2.1753	6651
PRLS Autism	**2.375**	33255
Cartesian GL Autism	2.3400	6651
Strong GL Autism	2.3563	6651

Comparison of our graph learning method with SCM and PRLS.

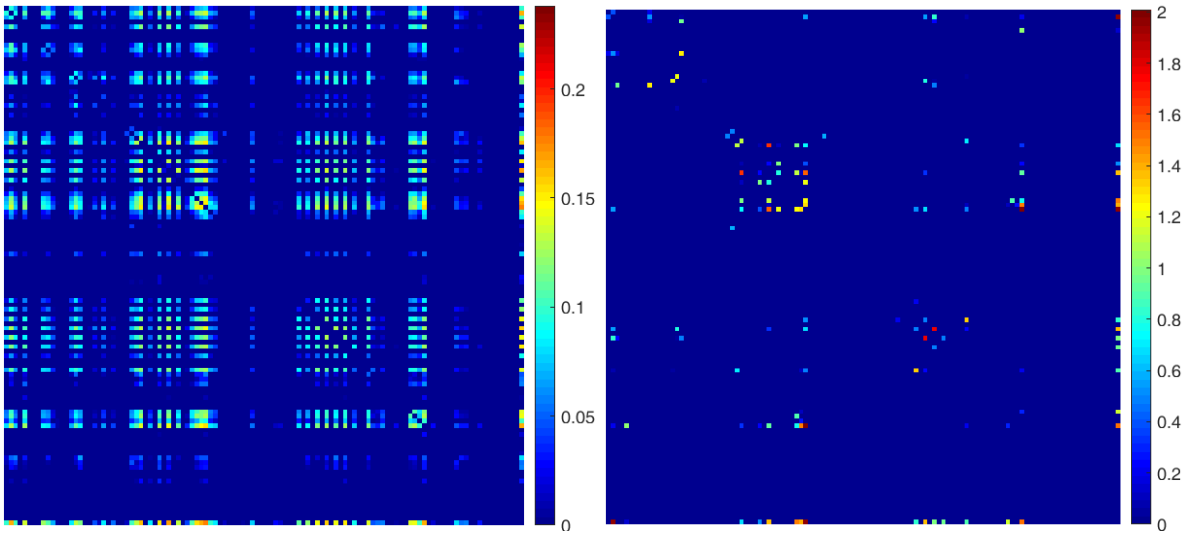

Figure 3.4: This figure shows the adjacency matrix of the spatial components learned for control (left) and autism (right) subjects with strong graph learning algorithms, respectively. The images reveal, in line with the existing literature, that control brain is much more connected than the autistic brain.

learn a Kronecker structured graph on this data using 120 randomly selected samples for training and the remaining 37 for testing. We choose Kronecker structured graph for this data because transposable models, i.e., models that learn a Kronecker structured data covariance, have been shown to work well for genomic data in the existing literature [61]. And as pointed out in Sec.3.4.1, Kronecker structured adjacency matrix corresponds to a Kronecker structured data covariance. For testing purposes, we follow a procedure similar to the previous subsections.

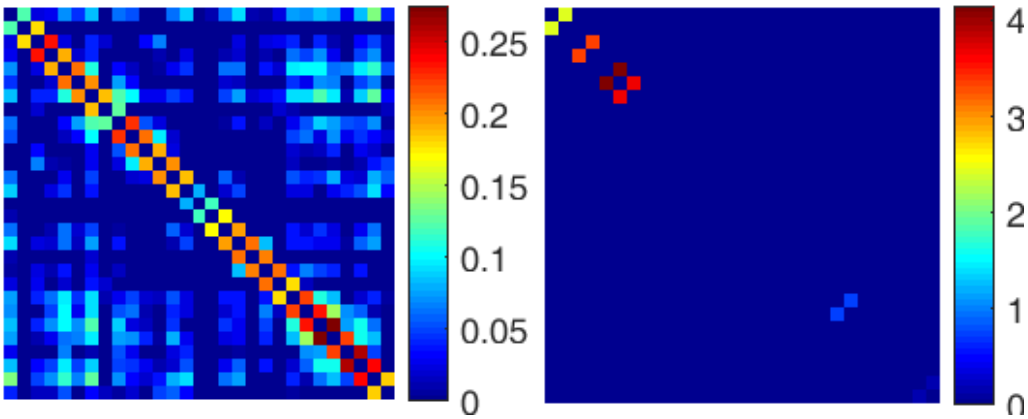

Figure 3.5: This figure shows the adjacency matrix of the temporal components learned for control (left) and autism (right) subjects with strong graph learning algorithms, respectively. The images reveal that control brains exhibit more temporal connections as compared to autistic brains. This is a new finding possible only by considering the spatiotemporal dynamics of the brain rather than just spatial connectivity analysis.

We compare our graph learning approach with SCM, PRLS and sparse covariance estimation (SEC) from [73]. Optimization parameters are manually tuned for best results for each method. We learn a graph through our method (and covariance through PRLS), with a Kronecker structure composed of two factor matrices of dimensions $n_1 = 21$ and $n_2 = 33$. We then use LMMSE estimator to predict 33 probe set measurements removed from the test data. PRLS, SEC and BPGL result in an improvement of $\mathbf{0.91347}$ dB, $\mathbf{0.93598}$ dB, and $\mathbf{1.0242}$ dB over SCM, respectively. This demonstrates that our method outperforms the state-of-the-art unstructured and structured sparse covariance estimation techniques, and provides a better model for real datasets.

3.7 Conclusion

In this chapter, we introduced a new linear formulation of the graph learning problem from graph signals. We demonstrated the performance of this new formulation with numerical experiments and derived bounds on its estimation performance. Based on the proposed formulation, we also posed the problem to learn product graphs from data. We devised

a block coordinate descent based algorithm for learning product graphs, and derived the associated error bounds for various product structures. Finally, we validated the performance characteristics and superior learning capabilities of our proposed method through numerical simulations on synthetic and real datasets.

3.8 Appendix

3.8.1 Proof of Theorem 8

Note that the current form of the objective function (3.11) can be expressed as:

$$\begin{aligned}
&\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\bar{\mathbf{x}}_m^T(\bigotimes_{[K_0]}\mathbf{W}_i)\mathbf{1}-\mathbf{x}_m^T(\bigotimes_{[K_0]}\mathbf{W}_i)\mathbf{x}_m\\
&=\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\langle\bar{\mathcal{X}}_m,\mathbb{1}\underset{[K_0]}{\times}\mathbf{W}_i\rangle-\langle\mathcal{X}_m,\mathcal{X}_m\underset{[K_0]}{\times}\mathbf{W}_i\rangle,
\end{aligned}\tag{3.22}$$

using the properties of the Kronecker product. Let us define the following: the tensor $\mathcal{Y}_m=\mathcal{X}_m\underset{[K_0]\setminus k}{\times}\mathbf{W}_i^{1/2}\underset{k}{\times}\mathbf{I}_k$, the matrix $\mathbf{S}_k=\frac{1}{M_0}\sum_{m=1}^{M_0}\mathcal{Y}_{m(k)}\mathcal{Y}_{m(k)}^T$, $\bar{\mathbf{d}}_k$ is a vector of degrees of the product adjacency matrix $\bigotimes_{[K_0]\setminus k}\mathbf{W}_i$, $\mathbf{d}_k$ is the vector of degrees of $\mathbf{W}_k$, $\bar{\mathbf{y}}_{mj}$ is the j-th column of $\bar{\mathcal{X}}_{m(k)}$, $\mathbf{y}_{mj}$ is the j-th column of $\mathcal{X}_{m(k)}$, and finally the matrix $\bar{\mathbf{S}}_k=\frac{1}{M_0}\sum_{m=1}^{M_0}\sum_{j=1}^{n/n_k}(\bar{\mathbf{d}}_k)_j\mathbf{y}_{mj}\mathbf{y}_{mj}^T$. Then we can further express the terms in (3.22) as:

$$\begin{aligned}
&\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\langle\mathcal{X}_m,\mathcal{X}_m\underset{[K_0]}{\times}\mathbf{W}_i\rangle\\
&=\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\big(\mathbf{W}_k\mathcal{X}_{m(k)}(\bigotimes_{[K_0]\setminus k}\mathbf{W}_i)\mathcal{X}_{m(k)}^T\big)\\
&=\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\big(\mathbf{W}_k\mathcal{Y}_{m(k)}\mathcal{Y}_{m(k)}^T\big)=\alpha\,\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big),
\end{aligned}\tag{3.23}$$

and,

$$
\begin{aligned}
&\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\langle \bar{\mathcal{X}}_m, \mathbb{1} \underset{[K_0]}{\times} \mathbf{W}_i\rangle \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\big(\mathbf{W}_k\bar{\mathcal{X}}_{m(k)}(\underset{[K_0]\setminus k}{\otimes}\mathbf{W}_i)\mathbb{1}_{(k)}^T\big) \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\big(\mathbf{W}_k\bar{\mathcal{X}}_{m(k)}\bar{\mathbf{d}}_k\mathbf{1}^T\big) \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\big(\mathbf{d}_k^T\bar{\mathcal{X}}_{m(k)}\bar{\mathbf{d}}_k\big) \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\big(\mathbf{d}_k^T[\bar{\mathbf{y}}_{m1}, \bar{\mathbf{y}}_{m2}, \cdots, \bar{\mathbf{y}}_{mn/n_k}]\bar{\mathbf{d}}_k\big) \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\sum_{j=1}^{n/n_k}\mathrm{trace}\big((\bar{\mathbf{d}}_k)_j\mathbf{y}_{mj}^T\mathbf{D}_k\mathbf{y}_{mj}\big) = \alpha\ \mathrm{trace}(\mathbf{D}_k\bar{\mathbf{S}}_k) \qquad (3.24)
\end{aligned}
$$

Moreover, for the terms in (3.23) and (3.24), the difference from their expected value takes the form of:

$$
\begin{aligned}
&\alpha\left|\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big) - \mathbb{E}_{\mathbf{x}}\big[\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big)\big]\right| \\
&= \alpha\left|\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big) - \mathrm{trace}\big(\mathbf{W}_k\mathbb{E}_{\mathbf{x}}[\mathbf{S}_k]\big)\right| \\
&= \alpha\left|\mathrm{trace}\big(\mathbf{W}_k(\mathbf{S}_k - \mathbb{E}_{\mathbf{x}}[\mathbf{S}_k])\big)\right| \\
&= \alpha\left|\sum_{i,j}\big(\mathbf{W}_k(\mathbf{S}_k - \mathbb{E}_{\mathbf{x}}[\mathbf{S}_k])\big)_{i,j}\right| \\
&\leq \alpha\max_{i,j}\Big|(\mathbf{S}_k - \mathbb{E}_{\mathbf{x}}[\mathbf{S}_k])_{i,j}\Big|\sum_{i,j}\big(\mathbf{W}_k\big)_{i,j} \\
&= \alpha n_k\max_{i,j}\Big|(\mathbf{S}_k - \mathbb{E}_{\mathbf{x}}[\mathbf{S}_k])_{i,j}\Big| \leq C_1\frac{n_k^2\log(n_k)}{nM_0}, \qquad (3.25)
\end{aligned}
$$

and

$$\begin{aligned}
&\alpha \left|\mathrm{trace}\big(\mathbf{D}_k\bar{\mathbf{S}}_k\big) - \mathbb{E}_{\mathbf{x}}\big[\mathrm{trace}\big(\mathbf{D}_k\bar{\mathbf{S}}_k\big)\big]\right| \\
&\leq \alpha \max_{i,j}\left|(\bar{\mathbf{S}}_k - \mathbb{E}_{\mathbf{x}}[\bar{\mathbf{S}}_k])_{i,j}\right| \sum_{i,j}\big(\mathbf{D}_k\big)_{i,j} \\
&= \alpha n_k \max_{i,j}\left|(\bar{\mathbf{S}}_k - \mathbb{E}_{\mathbf{x}}[\bar{\mathbf{S}}_k])_{i,j}\right| \leq C_2\frac{n_k^2\log(n_k)}{nM_0},
\end{aligned} \tag{3.26}$$

with probability for both inequalities exceeding $1 - 4n_k^2\Big[\exp\big(\frac{-nM_0}{2n_k}\big) + \exp\big(-(0.25 + \sqrt{\log n_k})^2\big)\Big]$; details in [74, Lemma B.1]. The last inequalities in both expressions follow from [74], and by choosing $\alpha \leq \sqrt{\frac{n_k\log(n_k)}{nM_0}}$. With these bounds, the error between the sample-based objective and the population-based objective can be upper-bounded as:

$$\begin{aligned}
&\alpha \Big|\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big) - \mathrm{trace}\big(\mathbf{D}_k\bar{\mathbf{S}}_k\big) \\
&- \mathbb{E}\big[\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big)\big] + \mathbb{E}\big[\mathrm{trace}\big(\mathbf{D}_k\bar{\mathbf{S}}_k\big)\big]\Big| \\
&\leq \alpha \Big|\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big) - \mathbb{E}\big[\mathrm{trace}\big(\mathbf{W}_k\mathbf{S}_k\big)\big]\Big| + \\
&+ \alpha \Big|\mathrm{trace}\big(\mathbf{D}_k\bar{\mathbf{S}}_k\big) - \mathbb{E}\big[\mathrm{trace}\big(\mathbf{D}_k\bar{\mathbf{S}}_k\big)\big]\Big| \leq C\frac{n_k^2\log(n_k)}{nM_0}
\end{aligned} \tag{3.27}$$

To derive the bound on the error between the factor estimate $\widehat{\mathbf{W}}_k$ and the true factor $\mathbf{W}_k$, let us first define the following convex function of $\mathbf{\Delta}$:

$$F_k(\mathbf{\Delta}) = \alpha\,\mathrm{trace}\big(\bar{\mathbf{S}}_k\mathrm{diag}(\mathbf{\Delta}\mathbf{1})\big) - \alpha\,\mathrm{trace}\big(\mathbf{S}_k\mathbf{\Delta}\big). \tag{3.28}$$

where $\mathbf{\Delta} = \mathbf{W}_k' - \mathbf{W}_k$, and $\mathrm{diag}(\mathbf{\Delta}\mathbf{1}) = \mathbf{D}_k' - \mathbf{D}_k$.

Now, we want to prove that $F_k(\mathbf{\Delta}) > 0$ for $\mathbf{\Delta} \in \mathbb{R}^{n_k\times n_k}$ with $\|\mathbf{\Delta}\|_F = \|\mathbf{W}_k' - \mathbf{W}_k\|_F = R\sqrt{\frac{n_k\log(n_k)}{nM_0}}$, for a constant $R > 0$. Consider F_k at $\widehat{\mathbf{\Delta}} = \widehat{\mathbf{W}}_k - \mathbf{W}_k$, which is the minima of $F_k(\mathbf{\Delta})$ because $\widehat{\mathbf{W}}_k$ is the minima of our factor-wise minimization in (3.19).

Then we have:

$$\begin{aligned} F_k(\widehat{\mathbf{\Delta}}) &= \alpha \mathrm{trace}\big(\widehat{\mathbf{D}}_k \bar{\mathbf{S}}_k - \widehat{\mathbf{W}}_k \mathbf{S}_k\big) - \alpha \mathrm{trace}\big(\mathbf{D}_k \bar{\mathbf{S}}_k - \mathbf{W}_k \mathbf{S}_k\big), \\ &\leq F_k(\mathbf{0}) = \alpha \mathrm{trace}\big(\mathbf{D}_k \bar{\mathbf{S}}_k - \mathbf{W}_k \mathbf{S}_k\big) - \\ &\qquad \alpha \mathrm{trace}\big(\mathbf{D}_k \bar{\mathbf{S}}_k - \mathbf{W}_k \mathbf{S}_k\big) = 0, \end{aligned} \tag{3.29}$$

If we can prove that $F_k(\mathbf{\Delta}) > 0$ for a $\mathbf{\Delta} \in \mathbb{R}^{n_k \times n_k}$ of certain norm, then since $F_k(\widehat{\mathbf{\Delta}}) < 0$, it must satisfy $\|\widehat{\mathbf{\Delta}}\|_F < R\sqrt{\frac{n_k \log(n_k)}{nM_0}}$. To see that $F_k > 0$ for $\mathbf{\Delta} \in \mathbb{R}^{n_k \times n_k}$ with the prescribed norm, first consider the following using the property of the trace of product of matrices [75]:

$$\begin{aligned} \mathrm{trace}\big(\bar{\mathbf{S}}_k \mathrm{diag}(\mathbf{\Delta}\mathbf{1})\big) &\geq \lambda_{n_k}(\bar{\mathbf{S}}_k) \mathrm{trace}\big(\mathrm{diag}(\mathbf{\Delta}\mathbf{1})\big) \\ &\geq \lambda_{n_k}(\bar{\mathbf{S}}_k) \|\mathrm{diag}(\mathbf{\Delta}\mathbf{1})\|_F \\ &= \lambda_{n_k}(\bar{\mathbf{S}}_k) \|\mathbf{\Delta}\mathbf{1}\|_F > 0 \end{aligned} \tag{3.30}$$

since $\|\mathbf{\Delta}\|_F \geq 0$, and where $\lambda_{n_k}(\bar{\mathbf{S}}_k)$ is the minimum eigenvalue of $\mathbf{S}_k$. Secondly, one can also see that:

$$\mathrm{trace}\big(\mathbf{S}_k \mathbf{\Delta}\big) \leq \lambda_1(\mathbf{S}_k) \mathrm{trace}\big(\mathbf{\Delta}\big) = 0, \tag{3.31}$$

where $\lambda_1(\mathbf{S}_k)$ is the largest eigenvalue of $\mathbf{S}_k$, and $\mathrm{trace}\big(\mathbf{\Delta}\big) = 0$ because of the adjacency constraints. Using the upper and lower bounds on the trace terms one can see that $F_k(\mathbf{\Delta}) > 0$ which completes the proof. ■

3.8.2 Proof of Theorem 9

Let us focus on the Cartesian objective function in (3.4):

$$\begin{aligned}
&\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathbf{x}_m^T(\bigoplus_{[K_0]}\mathbf{W}_i)\mathbf{x}_m \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\sum_{k=1}^{K_0}\mathbf{x}_m^T((\bigotimes_{[k-1]}\mathbf{I}_i)\otimes\mathbf{W}_k\otimes(\bigotimes_{[K_0]\setminus[k]}\mathbf{I}_j))\mathbf{x}_m \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\sum_{k=1}^{K_0}\langle\mathcal{X}_m,\mathcal{X}_m\underset{k}{\times}\mathbf{W}_k\rangle \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\sum_{k=1}^{K_0}\mathrm{trace}\big(\mathbf{W}_k\mathcal{X}_{m(k)}\mathcal{X}_{m(k)}^T\big) = \alpha\sum_{k=1}^{K_0}\mathrm{trace}\big(\mathbf{W}_k\mathbf{T}_k\big)
\end{aligned} \tag{3.32}$$

where the matrix $\mathbf{T}_k = \frac{1}{M_0}\sum_{m=1}^{M_0}\mathcal{X}_{m(k)}\mathcal{X}_{m(k)}^T$. Similar steps can be followed, along the lines of (3.24), to arrive at $\alpha/M_0\sum_{m=1}^{M_0}\mathbf{x}_m^T(\bigoplus_{[K_0]}\mathbf{D}_i)\mathbf{x}_m = \alpha\sum_{k=1}^{K_0}\mathrm{trace}\big(\mathbf{D}_k\bar{\mathbf{T}}_k\big)$. Thus, one can clearly see that the objective function can be expressed as a sum of terms each dependent on only one of the factor adjacency matrices $\mathbf{W}_k$. After this, one can follow the steps in Appendix 3.8.1 to obtain the final error bounds. ■

3.8.3 Proof of Theorem 10

Focusing again on the objective in (3.4), the terms involving only the k-th factor $\mathbf{L}_k$ can be expressed as:

$$\begin{aligned}
&\frac{\alpha}{M_0}\sum_{m=1}^{M_0}\sum_{j=0}^{K_0-1}\sum_{\mathbf{p}}^{\mathbf{P}(k,j)}\mathcal{X}_m{:}(\mathcal{X}_m\times_{\mathbf{p}}\mathbf{W}_{\mathbf{p}}\times_k\mathbf{W}_k) \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\sum_{j=0}^{K_0-1}\sum_{\mathbf{p}}^{\mathbf{P}(k,j)}\mathrm{trace}\big(\mathbf{W}_k\mathcal{X}_{m(k)}\mathbf{M}_{\mathbf{p}}\mathcal{X}_{m(k)}^T\big) \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\Big(\mathbf{W}_k\mathcal{X}_{m(k)}\Big[\sum_{j=0}^{K_0-1}\sum_{\mathbf{p}}^{\mathbf{P}(k,j)}\mathbf{M}_{\mathbf{p}}\Big]\mathcal{X}_{m(k)}^T\Big) \\
&= \frac{\alpha}{M_0}\sum_{m=1}^{M_0}\mathrm{trace}\big(\mathbf{W}_k\mathcal{X}_{m(k)}\mathbf{Q}_k\mathcal{X}_{m(k)}^T\big) = \alpha\,\mathrm{trace}\big(\mathbf{W}_k\mathbf{Z}_k\big),
\end{aligned} \tag{3.33}$$

where $\mathbf{p}$ denotes a column from matrix $\mathbf{P}(k,j)$, $\sum_{\mathbf{p}}^{\mathbf{P}(k,j)}$ denotes the summation over the columns of $\mathbf{P}(k,j)$, and the columns of $\mathbf{P}(k,j)$ are different combinations of indices given by $\left[\begin{smallmatrix}[1,\ldots,K_0]-[k]\\ j\end{smallmatrix}\right]$. Additionally, $\mathbf{M}_{\mathbf{p}}$ denotes a matrix of size $(n-n_k)\times(n-n_k)$ that contains an appropriate Kronecker product of identity matrices and factor adjacency matrices in accordance with the entries of the vector $\mathbf{p}$, and $\mathbf{Z}_k = \frac{1}{M_0}\sum_{m=1}^{M_0}\mathcal{X}_{m(k)}\mathbf{Q}_k\mathcal{X}_{m(k)}^T$. The remaining steps are similar to the proof of Theorem 9, and are thus omitted in the interest of space. ■

3.8.4 Proof of Theorem 11

To prove this lemma, one can follow along the lines of the proof for [76, Proposition 2.7.1]. Since each mode-wise problem is convex, the update for each mode-wise problem is guaranteed to converge to its minimum. Once block/mode-wise convergence to the minima is established, the convergence of every limit point to a stationary point is proven from [76, Proposition 2.7.1].

The work in [77] provides convergence guarantees and rates of convergence for block coordinate descent for multiconvex objectives. It can be trivially seen that each factor-wise problem (3.8) for learning factor graphs is strongly convex. The strong convexity of the factor-wise problems, in conjunction with [77, Theorem 2.9] (part 2), implies that Alg. 2 presented in this paper converges to its critical points at a linear rate. ■

CHAPTER 4
DISTRIBUTED RADAR IMAGING UNDER AMBIGUOUS ARRAY PARAMETERS

4.1 Introduction

Distributed radar imaging is an essential modern radar imaging technique as it enables high-resolution radar imaging through a large synthetic aperture. This large aperture is achieved by combining information from several geographical distributed radar platforms with small individual apertures. Such distributed radar arrays also facilitate a flexible platform that can be mobile, is tolerant to component failures, and admits low maintenance costs [78, 79, 80, 81]. However, such a setup with spatially distributed arrays is also prone to coherence issues caused by ambiguities in antennas' locations and complications in precise synchronization of antenna clocks.

To overcome these obstacles, two approaches have traditionally been the focus of attention in the literature. The first approach aims to design methods that are robust to the position and clock ambiguities by modeling them as errors that cannot be resolved [80, 82, 83]. This approach, aptly called incoherent imaging, typically results in low-resolution imaging and poor reconstruction performance.

A more successful approach, which has received a lot of attention over the years, is to resolve the ambiguities in distributed radar by modeling them as unknowns in the imaging problem. Most works under this approach have proposed schemes that compensate for both (or one) ambiguities by modeling them as unknown gain and phase errors in the acquired measurements [84, 85, 86, 87, 88, 89]. While this error model is valid under certain conditions, e.g., under clock mismatch the error is equivalent to a phase-only component that is linear in frequency, or under far-field imaging a position error is well approximated as

a time shift, for most real world scenarios the model is imprecise as investigated in [90]. Moreover, the resulting formulation is difficult to handle due to the non-linearities and required additional constraints that don't always translate to real situations; see [91, 92, 93] for details.

In lieu of this, recent work in distributed radar imaging has focused on developing precise formulations that model the ambiguities in a distributed setup in (i) the image domain for position errors in the antennas' locations [94, 90], and (ii) time domain for the clock mismatch between transmitting and receiving antennas [95]. These models have been shown to produce better reconstructions when compared to their imprecise counterparts that model all ambiguities with just a gain and phase error term. Our work in this dissertation is inspired by and is a continuation of these models to handle more general imaging scenarios. We first explicitly model the position errors in antennas' locations for the case when transmitting and receiving antennas are affected by different position errors, rather than the same error as posed in [94, 90]. We show how in this case, the problem can be posed as a blind deconvolution problem in the unknown position errors and the radar scene to be reconstructed. Afterwards, we revisit the convex formulation of the clock-mismatch problem posed in [95], and pose it as a blind deconvolution problem. Finally, we pose a general formulation that explicitly and jointly models the position ambiguity and the clock synchronization error to pose a multilinear blind deconvolution problem in all the unknowns. The essence of our work is precise formulation of ambiguities in distributed radar to achieve performance on par with coherent imaging. Our work hinges on the fact that proper use of knowledge about the structure in unknown variables leads to accurate models that outperform approximate models. We also devise algorithms for our proposed formulations and derive the associated error bounds on the estimated quantities. The identifiability conditions for the deconvolution problems are derived and the performance is verified through numerical simulations.

4.1.1 Organization

The rest of the chapter is organized as follows. We describe the general distributed radar imaging setup and formulate the imaging problem with ambiguities in Sec. 4.2. In Sec. 4.3 we pose the imaging problem under various ambiguities as blind deconvolution problems and provide a block coordinate descent based algorithm for solving the proposed deconvolution problems. Sec. 4.4 examines the theoretical guarantees for the blind deconvolution problems in terms of error of the reconstructed radar scene and the estimated ambiguities. Afterwards, we provide experimental verification of performance of the proposed algorithm in Sec. 4.5. and finally, Sec. 4.6 concludes the chapter by summarizing our results.

4.2 Problem formulation

We consider a two-dimensional radar scene of K targets in which the region of interest is divided into a spatial grid Ω containing $|\Omega| = N = N_x \times N_y$ points, where N_x and N_y represent the granularity of the grid in horizontal and vertical directions, respectively. We further assume the grid resolution is sufficiently fine and that there is only one reflector at each grid point. Let us also consider that the radar scene is being imaged with M antennas that could be situated inside or outside the scene grid. Furthermore, assume that the spatial locations of the antennas are known and are denoted by the set $\Gamma \subset \mathbb{R}^2$ with cardinality $|\Gamma| = M$. Without loss of generality, we assume that a subset of antennas act as both transmitters and receivers, while the others act only as receivers. To image the target scene, a time-domain pulse $p(t)$ with frequency spectrum $P(w)$ is transmitted by all transmitting antennas, where $w = 2\pi f$ is the angular frequency with f being the ordinary frequency in the bandwidth $\mathcal{B}$, where the bandwidth consist of $|\mathcal{B}| = F$ frequency components.

With a transmit antenna at position $\mathbf{r} \in \Gamma$, the frequency domain (also called measurement domain) signal received by a receive antenna at position $\mathbf{r}' \in \Gamma$ due to scattering of

the pulse by a target at location $\mathbf{l} \in \Omega$ is given by [96]:

$$Y(w, \mathbf{r}, \mathbf{r}', \mathbf{l}) = P(w)G(w, \mathbf{r}, \mathbf{r}', \mathbf{l})X(\mathbf{l}) + N(w), \tag{4.1}$$

where $X(\mathbf{l}) \in \mathbb{C}$ is the reflectivity of the radar scene at location $\mathbf{l}$, $N(w)$ is the measurement noise, and $G(w, \mathbf{r}, \mathbf{r}', \mathbf{l})$ is the pulse propagation characterized by:

$$G(w, \mathbf{r}, \mathbf{r}', \mathbf{l}) = a(\mathbf{r}, \mathbf{r}', \mathbf{l})e^{-iw\frac{\|\mathbf{r}-\mathbf{l}\|_2+\|\mathbf{r}'-\mathbf{l}\|_2}{c}}. \tag{4.2}$$

In (4.2), $a(\mathbf{r}, \mathbf{r}', \mathbf{l})$ term denotes magnitude attenuation whereas $e^{-iw\frac{\|\mathbf{r}-\mathbf{l}\|_2+\|\mathbf{r}'-\mathbf{l}\|_2}{c}}$ describes the phase change due to the transmission delay, and c denotes the speed of light.

Assuming no shadowing and no multiple reflections, the received signal for each transmitter-receiver pair is a sum of (4.1) for all locations $\mathbf{l} \in \Omega$ where a reflector is present. Denoting the vectorized scene reflectivity by $\mathbf{x} \in \mathbb{C}^N$, where a zero entry in $\mathbf{x}$ represents absence of a reflector at the respective grid point, one can express the received signal $\mathbf{y}(\mathbf{r}, \mathbf{r}') \in \mathbb{C}^F$ for a particular transmitter-receiver pair and at all frequencies w as follows:

$$\mathbf{y}(\mathbf{r}, \mathbf{r}') = \mathbf{A}(\mathbf{r}, \mathbf{r}')\mathbf{x}(\mathbf{l}) + \mathbf{n}(\mathbf{r}, \mathbf{r}'), \tag{4.3}$$

where $\mathbf{A}(\mathbf{r}, \mathbf{r}') \in \mathbb{C}^{F\times N}$ is dependent on $P(w)$ and $G(w, \mathbf{r}, \mathbf{r}', \mathbf{l})$, and denotes the radar imaging operator for the transmitter at position $\mathbf{r}$ and the receiver at position $\mathbf{r}'$. Moreover, $\mathbf{n}(\mathbf{r}, \mathbf{r}')$ denotes the measurement noise for the particular transmitter-receiver pair. An example of such distributed setup with 32 antennas grouped into 4 antenna arrays (each with 8 antennas) is shown in Fig. 4.1.

The distributed radar setup described in (4.3) and depicted in Fig. 4.1 is primarily affected by two type of ambiguities: (i) time ambiguity caused by unsynchronized clocks between transmitting and receiving antennas, and (ii) position ambiguity as a consequence of access to imprecise locations of the antennas comprising the distributed setup. In the

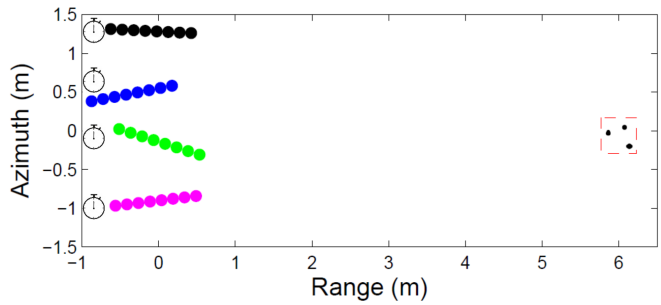

Figure 4.1: An ideal distributed radar setup. The colored dots represent the error-free positions of the antennas imaging the setup and the synchronized clocks represent the clocks for each antenna array. The target scene in consideration is represented by the red box with three targets in the scene.

following discussion we formalize the forward models when either one or both of these ambiguities are present in the imaging setup.

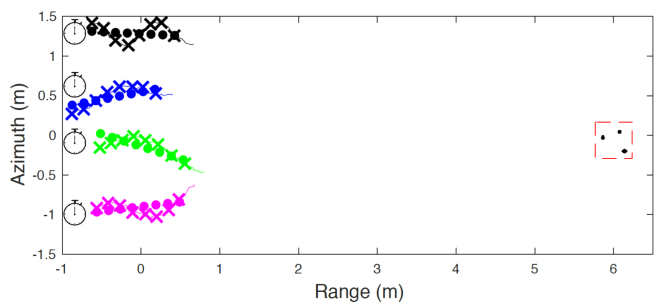

Figure 4.2: A distributed radar setup with synchronized clocks but position ambiguity. The crosses represent the actual positions of the antennas imaging the setup whereas the dots represent the erroneous assumed positions. As before, the target scene in consideration is represented by the red box with three targets in the scene.

4.2.1 Image-domain convolution model for position ambiguities

We first focus on the forward model for the distributed radar setup with erroneous antenna positions. Let us consider a transmitter-receiver pair with true transmitter and receiver positions denoted by $\mathbf{r}$ and $\mathbf{r}'$, respectively. Furthermore, let the erroneous antenna positions of the transmitter and receiver be denoted by $\widetilde{\mathbf{r}} = \mathbf{r} + \mathbf{e}$ and $\widetilde{\mathbf{r}}' = \mathbf{r}' + \mathbf{e}'$, respectively. Instead of modeling these position errors in the measurement (frequency) domain, as has

been done in the majority of existing works [84, 85, 86, 87, 88, 89, 91, 92, 93, 97, 94], we propose to model them in the image (spatial) domain. The reason being that modeling the position ambiguities as unknown complex quantities in the measurement domain amounts to approximating them as time-domain shift operators. As shown by recent works [94, 90], this approximation for position errors only holds in the far-field regime.

To handle these position errors in a precise manner, each antenna position ambiguity is modeled as an image-domain unknown shift kernel. We present our proposed image-domain convolution model for the distributed radar setup with position ambiguities in the following proposition:

Proposition 1. *Let* $\widetilde{\mathbf{y}} = \widetilde{\mathbf{A}}\mathbf{x}$ *be the observation for a transmitter-receiver pair with erroneous positions given by* $\widetilde{\mathbf{r}}$ *and* $\widetilde{\mathbf{r}}'$*. Then the equivalent image-domain convolution model can be expressed entrywise as:*

$$\widetilde{\mathbf{y}}(w) = \mathbf{A}'_w\Big(\big\{(\mathbf{A}_w * \mathbf{h})\,\mathbf{x}\big\} * \mathbf{g}\Big), \tag{4.4}$$

where $\mathbf{h}$ *and* $\mathbf{g}$ *represent the spatial shift kernels in the image-domain caused by the position errors of the transmitter and the receiver, respectively.*

Proof. Let us first express the observation vector $\widetilde{\mathbf{y}}$ entrywise as:

$$\begin{aligned}
\widetilde{\mathbf{y}}(w) &= \sum_{\mathbf{l}\in\Omega} e^{-iw\frac{\|\widetilde{\mathbf{r}}-\mathbf{l}\|_2+\|\widetilde{\mathbf{r}}'-\mathbf{l}\|_2}{c}}\mathbf{x}(\mathbf{l}) \\
&= \sum_{\mathbf{l}\in\Omega} e^{-iw\frac{\|\widetilde{\mathbf{r}}'-\mathbf{l}\|_2}{c}} e^{-iw\frac{\|\widetilde{\mathbf{r}}-\mathbf{l}\|_2}{c}}\mathbf{x}(\mathbf{l}) \\
&= \sum_{\mathbf{l}\in\Omega} e^{-iw\frac{\|\mathbf{r}'-(\mathbf{l}-\mathbf{e}')\|_2}{c}} e^{-iw\frac{\|\mathbf{r}-(\mathbf{l}-\mathbf{e})\|_2}{c}}\mathbf{x}(\mathbf{l}) \\
&= \mathbf{A}'_w(\mathbf{l}-\mathbf{e}')\;\mathbf{A}_w(\mathbf{l}-\mathbf{e})\;\mathbf{x}(\mathbf{l})
\end{aligned} \tag{4.5}$$

where $\mathbf{A}_w(\mathbf{l}-\mathbf{e})$ is a diagonal matrix with entries $e^{-iw\frac{\|\mathbf{r}-(\mathbf{l}-\mathbf{e})\|_2}{c}}$ for $\mathbf{l} \in \Omega$ on the diagonal, and $\mathbf{A}'_w(\mathbf{l}-\mathbf{e}')$ is a row matrix (matrix with just one row) with each entry given by

$e^{-iw\frac{\|\mathbf{r}'-(\mathbf{l}-\mathbf{e}')\|_2}{c}}$ for $\mathbf{l} \in \Omega$. Let's define $\mathbf{l}_e = \mathbf{l} - \mathbf{e}$. This implies that:

$$\begin{aligned}
\widetilde{\mathbf{y}}(w) &= \mathbf{A}'_w(\mathbf{l}-\mathbf{e}')\ \mathbf{A}_w(\mathbf{l}-\mathbf{e})\ \mathbf{x}(\mathbf{l}) \\
&= \mathbf{A}'_w(\mathbf{l}_e+\mathbf{e}-\mathbf{e}')\ \mathbf{A}_w(\mathbf{l}_e)\ \mathbf{x}(\mathbf{l}_e+\mathbf{e}) \\
&= \mathbf{A}'_w(\mathbf{l}_e-(\mathbf{e}'-\mathbf{e}))\ \mathbf{A}_w(\mathbf{l}_e)\ (\mathbf{x}(\mathbf{l}_e)*\delta(\mathbf{e})). \qquad (4.6)
\end{aligned}$$

Now, let us define $\overline{\mathbf{x}}(\mathbf{l}_e) = \mathbf{A}_w(\mathbf{l}_e)\ (\mathbf{x}(\mathbf{l}_e)*\delta(\mathbf{e}))$ and $\bar{\mathbf{l}} = \mathbf{l}_e - \overline{\mathbf{e}}$, where $\overline{\mathbf{e}} = \mathbf{e}' - \mathbf{e}$. Then we can express the above equation as:

$$\begin{aligned}
\widetilde{\mathbf{y}}(w) &= \mathbf{A}'_w(\mathbf{l}_e-(\mathbf{e}'-\mathbf{e}))\ \mathbf{A}_w(\mathbf{l}_e)\ (\mathbf{x}(\mathbf{l}_e)*\delta(\mathbf{e})) \\
&= \mathbf{A}'_w(\mathbf{l}_e-\overline{\mathbf{e}})\ \overline{\mathbf{x}}(\mathbf{l}_e) \\
&= \mathbf{A}'_w(\bar{\mathbf{l}})\ \overline{\mathbf{x}}(\bar{\mathbf{l}}+\overline{\mathbf{e}}) \\
&= \mathbf{A}'_w(\bar{\mathbf{l}})\ (\overline{\mathbf{x}}(\bar{\mathbf{l}})*\delta(\overline{\mathbf{e}})) \\
&= \mathbf{A}'_w(\bar{\mathbf{l}})\ \Big(\big(\mathbf{A}_w(\bar{\mathbf{l}})\ (\mathbf{x}(\bar{\mathbf{l}})*\delta(\mathbf{e}))\big)*\delta(\overline{\mathbf{e}})\Big). \qquad (4.7)
\end{aligned}$$

For further simplification, recognize that $\delta(\overline{\mathbf{e}}) = \delta(\mathbf{e}')*\delta(-\mathbf{e}')$, and moving the convolution with $\delta(-\mathbf{e})$ inside the brackets leads to:

$$\begin{aligned}
\widetilde{\mathbf{y}}(w) &= \mathbf{A}'_w(\bar{\mathbf{l}})\ \Big(\big(\mathbf{A}_w(\bar{\mathbf{l}})\ (\mathbf{x}(\bar{\mathbf{l}})*\delta(\mathbf{e}))\big)*\delta(\overline{\mathbf{e}})\Big) \\
&= \mathbf{A}'_w(\bar{\mathbf{l}})\ \Big(\big(\mathbf{A}_w(\bar{\mathbf{l}})\ (\mathbf{x}(\bar{\mathbf{l}})*\delta(\mathbf{e}))\big)*\delta(\mathbf{e}')*\delta(-\mathbf{e})\Big) \\
&= \mathbf{A}'_w(\bar{\mathbf{l}})\ \Big(\big\{(\mathbf{A}_w(\bar{\mathbf{l}})*\delta(-\mathbf{e}))\ \mathbf{x}(\bar{\mathbf{l}})\big\}*\delta(\mathbf{e}')\Big), \qquad (4.8)
\end{aligned}$$

which gives us the desired model by assuming $\mathbf{h} = \delta(-\mathbf{e})$ and $\mathbf{g} = \delta(\mathbf{e}')$. ■

A depiction of the imaging setup and the interpretation of the model in (4.4) can be seen in Fig. 4.2 and Fig. 4.3, respectively. Our objective under this image-domain convolution model is to recover the radar scene $\mathbf{x}$ as well as the antenna position ambiguities, $\mathbf{h}$ and $\mathbf{g}$.

Remark 9. Earlier work in [90] investigated image-domain modeling of position errors for

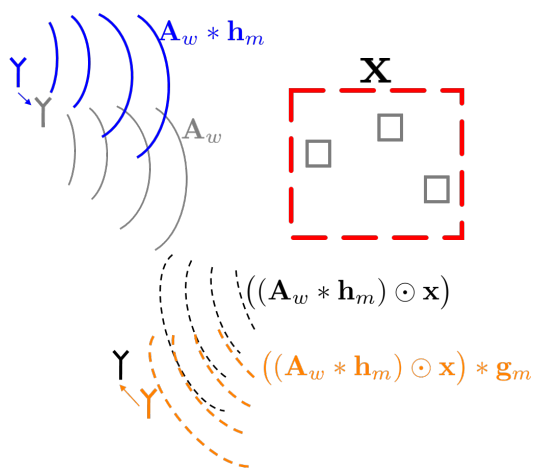

Figure 4.3: Interpretation of the model in (4.4) for distributed radar setup with position ambiguity. The red box denotes the target radar scene. The actual and assumed transmitting antenna positions are represented by the blue and gray antennas, respectively. Similarly, the actual assumed receiving antenna positions are, respectively, denoted by orange and black antennas. The corresponding incident fields (solid lines) and reflected fields (dotted lines) follow the same color notation.

the case of collocated transmitter-receiver pairs. In this case, both antennas suffer from the same position error and the model can be simplified. The model in Proposition 1 is more general and subsumes the model in [90]. Similar to the model in [90], if the transmitter and receiver antennas are both affected by the same position error, then $\overline{\mathbf{e}} = \mathbf{e}' - \mathbf{e} = \mathbf{0}$ in (4.7). In this scenario, the image-domain model gets simplified to $\widetilde{\mathbf{y}} = \mathbf{A}(\mathbf{x} * \mathbf{h})$, which is the model proposed in [90].

4.2.2 Measurement-domain model for clock mismatch

The second type of ambiguity that arises in the distributed radar setup is the time ambiguity due to unsynchronized clocks between transmitting and receiving antennas. The pairwise clock mismatch for each transmitter-receiver pair causes a time drift, which can be represented as a convolution with a time shift in the time-domain measurements. In the measurement domain, this time shift can be represented as phase-only component [95]. Denoting the time mismatch affecting a particular transmitter-receiver pair by $\mathbf{z}$, the received

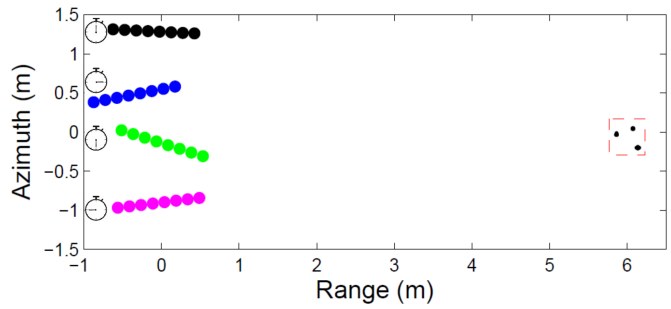

Figure 4.4: A distributed radar setup with correct antenna locations but unsynchronized clocks. The colored dots represent the error-free positions of the antennas imaging the setup. The clocks across the antenna arrays are not synchronized which results in the time ambiguity. The target scene in consideration is represented by the red box with three targets in the scene.

frequency domain signal for the pair (4.3) can be rewritten as:

$$\widetilde{\mathbf{y}} = \mathbf{D}_{\overline{\mathbf{z}}}\mathbf{A}\mathbf{x} + \mathbf{n}, \tag{4.9}$$

where $\mathbf{A}$ succinctly represents the measurement matrix without position errors, $\overline{\mathbf{z}} = \mathbf{F}\mathbf{z}$, $\mathbf{F}$ is the Fourier transform matrix, and $\mathbf{D}_{\overline{\mathbf{z}}}$ is a diagonal matrix with the Fourier transform of $\mathbf{z}$ at its diagonal. The goal under this model is to recover the radar scene $\mathbf{x}$ and the unknown time mismatch $\mathbf{z}$. A depiction of the imaging setup and an interpretation of the model in (4.9) are shown in Fig. 4.4 and Fig. 4.5, respectively.

This model and its variants have a rich history in distributed radar and have been used in previous studies for modeling the combined effect of time and position ambiguities [84, 85, 86, 87, 88, 89, 94]. However, as noted in Sec. 4.2.1, this model is exact only when either the measurements are being affected by just the time ambiguity, or by position ambiguity in the far field.

4.2.3 Generalized model for both position and time ambiguities

In light of the preceding discussion, we can now formulate a generalized model for distributed radar that incorporates both time and position ambiguities in the forward model.

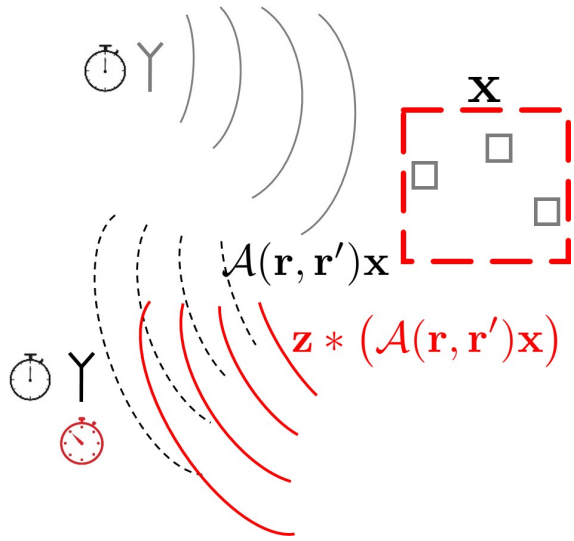

Figure 4.5: Interpretation of the model in (4.9) for distributed radar setup with time ambiguity. The red box denotes the target radar scene. The transmitting antenna position is represented by the gray antenna, whereas the receiving antenna position is denoted by the black antenna. The incident field is represented with solid gray lines. The reflected fields (dotted lines) in the absence and presence of time mismatch are shown in black and red, respectively.

We introduce the proposed forward model for a particular transmitter-receiver pair, affected by both position and time ambiguities, in the following proposition.

Proposition 2. Let $\widetilde{\mathbf{y}} = \widetilde{\mathbf{A}}\mathbf{x}$ *be the observation for a transmitter-receiver pair with erroneous positions given by* $\widetilde{\mathbf{r}}$ *and* $\widetilde{\mathbf{r}}'$*, as well as a time mismatch given by* $\mathbf{z}$*. Then, the equivalent image-domain convolution model can be expressed entrywise as:*

$$\widetilde{\mathbf{y}}(w) = \mathbf{D}_{\widetilde{\mathbf{z}}}(w)\mathbf{A}'_w\Big(\big\{(\mathbf{A}_w * \mathbf{h})\,\mathbf{x}\big\} * \mathbf{g}\Big), \tag{4.10}$$

where $\mathbf{D}_{\widetilde{\mathbf{z}}}(w)$ *denotes the diagonal entry of* $\mathbf{D}_{\widetilde{\mathbf{z}}}$ *indexed by* w*.*

This proposition follows directly from Proposition 1 and (4.9). The forward model aims to jointly recover the radar scene $\mathbf{x}$, the time ambiguity $\mathbf{z}$, and the antenna position ambiguities $\mathbf{h}$ and $\mathbf{g}$. The novelty of this proposed forward model is the explicit separate representation of the position and time ambiguities. This separation of the two ambiguities

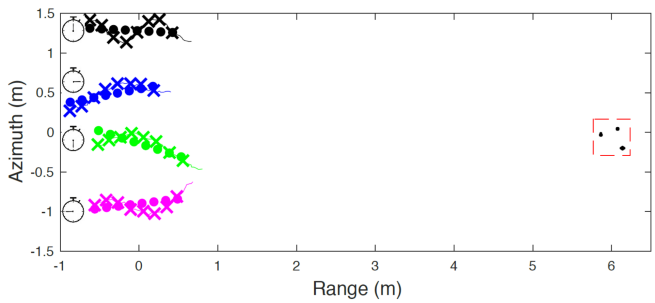

Figure 4.6: A distributed radar setup with both position and time ambiguity. The crosses represent the actual positions of the antennas imaging the setup whereas the dots represent the erroneous assumed positions. The clocks across the antenna arrays are not synchronized which results in the time ambiguity. The target scene in consideration is represented by the red box with three targets in the scene.

allows for a precise characterization of the distributed radar setup and for a better utilization of the known properties of both ambiguities, which produces a better reconstruction of the radar scene.

4.3 Blind deconvolution for ambiguous distributed radar

In this section we first pose the problems proposed in Sec. 4.2 as blind deconvolution problems, followed by the algorithms to arrive at their solutions.

4.3.1 Blind deconvolution for position ambiguities

We first examine the case of distributed radar with just position ambiguities. To recover the radar scene and the position errors from the model in Prop. 1, a simple minimization over $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{g}$ would be insufficient as the problem would be highly ill-posed. Therefore, one needs to incorporate prior knowledge about the radar scene and the position ambiguities to formulate a well-posed problem. To this end, we first realize that the radar scene $\mathbf{x}$ is sparse and piecewise smooth. Moreover, the shift kernels modeling the position ambiguities are non negative 1-sparse two-dimensional shift operators. We include these properties in our model through a fused Lasso [98] penalty for the radar scene, whereas a ℓ_1-norm penalty for

the shift kernels while constraining them to be on the standard simplex (to enforce 1-sparse solutions). The resulting overall optimization problem for a total of M observations, where each observation is an interaction between one transmitter-receiver pair, is as follows:

$$\begin{aligned} \min_{\substack{\mathbf{x}\in\mathbb{C}^N,\\ \mathbf{h}_m\in\mathbb{R}_+^{N_h},\\ \mathbf{g}_m\in\mathbb{R}_+^{N_g}}} & \sum_{m=1}^{M} \|\widetilde{\mathbf{y}}_m - \mathbf{A}_m\mathbf{x}\|_2^2 + \alpha(\|\mathbf{h}_m\|_1 + \|\mathbf{g}_m\|_1) \\ \text{s.t.} \quad & \mathbf{R}_{\mathbf{x}}(\mathbf{x}) \leq \tau, \;\; \mathbf{1}^T\mathbf{h}_m = 1, \;\; \mathbf{1}^T\mathbf{g}_m = 1, \end{aligned} \tag{4.11}$$

where $\mathbf{h}_m$ and $\mathbf{g}_m$ represent the shift kernels of the m-th observation, $\mathbf{A}_m$ succinctly represents the shift kernel dependent pairwise measurement matrix for the m-th transmitter-receiver pair from (4.4), $\mathbf{R}_{\mathbf{x}}(\mathbf{x}) = \|\mathbf{x}\|_1 + \gamma\|\mathbf{x}\|_{TV}$ is the fused Lasso regularization, and $\|\cdot\|_{TV}$ is the well-known total variation regularization to promote piecewise smoothness [98]. The parameters α and τ are regularization parameters that control the sparsity and piecewise-smoothness of the shift kernels and the radar scene, respectively. This optimization problem is clearly nonconvex due to the convolution of the unknown variables, and is usually referred to as blind deconvolution (trilinear blind deconvolution in our case) in the literature [97, 92, 91, 90].

4.3.2 Blind deconvolution for clock mismatch

Next we focus on distributed radar under clock mismatch. Similar to Sec.4.3.1, we need additional constraints on the unknown variables to arrive at a well-posed optimization problem. As before, we regularize the radar scene with a fused Lasso penalty, and for the unknown time shift we use a ℓ_1-norm penalty for sparsity with a unit-sum constraint over

non-negative reals. The overall optimization problem then becomes:

$$\min_{\substack{\mathbf{x}\in\mathbb{C}^N,\\ \mathbf{z}_m\in\mathbb{R}_+^{N_z}}} \sum_{m=1}^{M} \|\widetilde{\mathbf{y}}_m - \mathbf{A}_m\mathbf{x}\|_2^2 + \beta\|\mathbf{z}_m\|_1$$
$$\text{s.t.} \quad \mathbf{R}_{\mathbf{x}}(\mathbf{x}) \leq \tau, \;\; \mathbf{1}^T\mathbf{z}_m = 1, \tag{4.12}$$

where $\overline{\mathbf{z}}_m = \mathbf{F}\mathbf{z}_m$, $\mathbf{z}_m$ represents the clock drift for the m-th transmitter-receiver pair, and $\mathbf{A}_m$ succinctly represents the clock drift dependent pairwise measurement matrix for the m-th transmitter-receiver pair from (4.9). This problem is also a nonconvex bilinear blind deconvolution problem similar to the problem for position ambiguities.

4.3.3 Blind deconvolution for the generalized model

We now present the optimization problem for the generalized model. Since the generalized model considers both position and synchronization errors, the associated optimization problem incorporates the knowledge for the radar scene, the spatial shift kernels, and the clock drift. We can express the overall problem as follows:

$$\min_{\substack{\mathbf{x}\in\mathbb{C}^N,\mathbf{z}_m\in\mathbb{R}_+^{N_z},\\ \mathbf{h}_m\in\mathbb{R}_+^{N_h},\mathbf{g}_m\in\mathbb{R}_+^{N_g}}} \sum_{m=1}^{M} \left\|\widetilde{\mathbf{y}}_m - \mathbf{D}_{\overline{\mathbf{z}}_m}\mathbf{A}_m\mathbf{x}\right\|_2^2 + \beta\|\mathbf{z}_m\|_1 + \alpha(\|\mathbf{h}_m\|_1 + \|\mathbf{g}_m\|_1)$$
$$\text{s.t.} \quad \mathbf{R}_{\mathbf{x}}(\mathbf{x}) \leq \tau, \; \mathbf{1}^T\mathbf{z}_m = 1, \mathbf{1}^T\mathbf{h}_m = 1, \; \mathbf{1}^T\mathbf{g}_m = 1, \tag{4.13}$$

which is a quadrilinear deconvolution problem and where $\mathbf{A}_m$ succinctly represents the clock drift dependent pairwise measurement matrix for the m-th transmitter-receiver pair from (4.10).

4.3.4 Block coordinate descent for blind deconvolution

Here we describe an alternating-minimization based block coordinate descent (BCD) algorithm for the optimization problems posed in the preceding subsections. As noted ear-

lier, the optimization problems are multilinear blind deconvolution problems. This means by fixing all but one unknown variable, the resulting problem is convex in that variable. Therefore, one way to arrive at a stationary point of the overall problem is to minimize individually over each variable while keeping the other variables fixed (by using their current estimates) [77]. The proposed algorithm for solving the generalized model with both position and synchronization ambiguities, named **Blo**ck coordinate descent for **G**eneralized blind **D**econvolution (**BloGD**), is outlined in Algorithm 3. For solving either position or synchronization error, the algorithm can be modified and different steps in the algorithm can be skipped depending on what one needs to estimate. In Algorithm 3, the operators

Algorithm 3: : BloGD—Block coordinate descent for **G**eneralized blind **D**econvolution

Input: Observations $\{\widetilde{\mathbf{y}}_m\}_{m=1}^M$
Initialize: initial radar scene estimate $\mathbf{x}^0$, initial ambiguity estimates $\{\mathbf{z}_m^0\}_{m=1}^M$,$\{\mathbf{h}_m^0\}_{m=1}^M$,$\{\mathbf{g}_m^0\}_{m=1}^M$, maximum inner iterations T, and parameters τ, α, β

repeat
$n \leftarrow n+1$
Estimate radar scene:
Update $\mathcal{A}_{\mathbf{x},m}$ with $\mathbf{z}_m^{n-1}$, $\mathbf{h}_m^{n-1}$, $\mathbf{g}_m^{n-1}$ for all m
Update τ according to [99]
$\mathbf{x}^n \leftarrow \text{FPGD}(\{\mathcal{A}_{\mathbf{x},m}\}_{m=1}^M, \{\widetilde{\mathbf{y}}_m\}_{m=1}^M, \mathbf{x}^{n-1}, \tau)$
Estimate position error: (*skip if only clock mismatch*)
for $m = 1$ to M
Update $\mathcal{A}_{\mathbf{h},m}$ with $\mathbf{x}^n$, $\mathbf{z}_m^{n-1}$, $\mathbf{g}_m^{n-1}$
$\mathbf{h}_m^n \leftarrow \text{FISTA}(\mathcal{A}_{\mathbf{h},m}, \widetilde{\mathbf{y}}_m, \mathbf{h}_m^{n-1}, T, \alpha)$
$\mathbf{h}_m^n \leftarrow \mathcal{P}_\infty(\mathbf{h}_m^n)$
for $m = 1$ to M
Update $\mathcal{A}_{\mathbf{g},m}$ with $\mathbf{x}^n$, $\mathbf{z}_m^{n-1}$, $\mathbf{h}_m^n$
$\mathbf{g}_m^n \leftarrow \text{FISTA}(\mathcal{A}_{\mathbf{g},m}, \widetilde{\mathbf{y}}_m, \mathbf{g}_m^{n-1}, T, \alpha)$
$\mathbf{g}_m^n \leftarrow \mathcal{P}_\infty(\mathbf{g}_m^n)$
Estimate clock drift: (*skip if only position errors*)
for $m = 1$ to M
Update $\mathcal{A}_{\mathbf{z},m}$ with $\mathbf{x}^n$, $\mathbf{h}_m^n$, $\mathbf{g}_m^n$
$\mathbf{z}_m^n \leftarrow \text{FISTA}(\mathcal{A}_{\mathbf{z},m}, \widetilde{\mathbf{y}}_m, \mathbf{z}_m^{n-1}, T, \beta)$
$\mathbf{z}_m^n \leftarrow \mathcal{P}_\infty(\mathbf{z}_m^n)$
until: stopping criterion
Output: Radar scene estimate $\mathbf{x}^n$.

Algorithm 4: : FPGD for updating radar scene $\mathbf{x}$

Input: Observations $\{\widetilde{\mathbf{y}}_m\}_{m=1}^M$, forward model operators $\{\mathcal{A}_{\mathbf{x},m}\}_{m=1}^M$, previous radar scene estimate $\mathbf{x}^{n-1}$, maximum iterations T, parameter τ

Initialize: $q^0 \leftarrow 1$, $\mathbf{u}^0 \leftarrow \mathbf{s}^0 \leftarrow \mathbf{x}^{n-1}$, $\gamma \leftarrow$ inverse of max eigenvalue of $\sum_{m=1}^{M} \mathcal{A}_{\mathbf{h},m}^H \mathcal{A}_{\mathbf{h},m}$

for $t = 1$ to T

$\mathbf{u}^t \leftarrow \mathcal{P}_{\mathbf{R_x}}\Big(\mathbf{s}^{t-1} + \gamma \sum_{m=1}^{M} \mathcal{A}_{\mathbf{h},m}^H(\widetilde{\mathbf{y}}_m - \mathcal{A}_{\mathbf{h},m}\mathbf{s}^{t-1}), \tau\Big)$

$q^t \leftarrow \frac{1+\sqrt{1+4(q^{t-1})^2}}{2}$

$\mathbf{s}^t \leftarrow \mathbf{u}^t + \frac{q^{t-1}-1}{q^t}(\mathbf{u}^t - \mathbf{u}^{t-1})$

Output: Radar scene estimate $\mathbf{x}^n \leftarrow \mathbf{s}^t$.

Algorithm 5: : FISTA for updating ambiguity $\mathbf{h}_m$

Input: Observation $\widetilde{\mathbf{y}}_m$, forward model operator $\mathcal{A}_{\mathbf{h},m}$, previous ambiguity estimate $\mathbf{h}_m^{n-1}$, maximum iterations T, parameter α

Initialize: $q^0 \leftarrow 1$, $\mathbf{u}^0 \leftarrow \mathbf{s}^0 \leftarrow \mathbf{h}_m^{n-1}$, $\gamma \leftarrow$ inverse of max eigenvalue of $\mathcal{A}_{\mathbf{h},m}^H \mathcal{A}_{\mathbf{h},m}$

for $t = 1$ to T

$\mathbf{u}^t \leftarrow \mathcal{T}_+\Big(\mathbf{s}^{t-1} + \gamma \mathcal{A}_{\mathbf{h},m}^H(\widetilde{\mathbf{y}}_m - \mathcal{A}_{\mathbf{h},m}\mathbf{s}^{t-1}), \gamma\alpha\Big)$

$\mathbf{u}^t \leftarrow \mathbf{u}^t/(\mathbf{1}^T\mathbf{u}^t)$

$q^t \leftarrow \frac{1+\sqrt{1+4(q^{t-1})^2}}{2}$

$\mathbf{s}^t \leftarrow \mathbf{u}^t + \frac{q^{t-1}-1}{q^t}(\mathbf{u}^t - \mathbf{u}^{t-1})$

Output: Ambiguity estimate $\mathbf{h}_m^n \leftarrow \mathbf{s}^t$.

$\mathcal{A}_{\mathbf{x},m}$, $\mathcal{A}_{\mathbf{h},m}$, and $\mathcal{A}_{\mathbf{z},m}$ represent the variable specific forward models for the m-th observation when all the other unknowns are replaced by their estimates. Moreover, $\mathcal{P}_\infty(\cdot)$ represents the projection onto the set of valid shift operators by replacing the largest entry in its argument by one while making all the other ones zero.

It can be seen that the objective functions in (4.11),(4.12) and (4.13) are separable in $\mathbf{h}_m$, $\mathbf{g}_m$ and $\mathbf{z}_m$ for all m. This means that the problem for updating each of these variable for a particular m reduces to standard non-negative sparse recovery problem, which can be efficiently solved through the fast iterative shrinkage/thresholding algorithm (FISTA) [99]. As for updating the radar scene $\mathbf{x}$, the problem is similar to standard sparse recovery but with a fused Lasso penalty instead of the Lasso penalty. For solving this fused Lasso prob-

lem for $\mathbf{x}$, we use the FISTA-inspired fast proximal gradient descent (FPGD) algorithm devised in [94] which replaces the proximal gradient step with projected gradient step. These algorithms are outlined in Algorithm4 and Algorithm5, where $\mathcal{T}_+(\cdot, \gamma\alpha)$ represents the well-known elementwise nonnegative soft-thresholding operator induced by the proximal shrinkage for the ℓ_1 regularization [99, 94] and $\mathcal{P}_{\mathbf{R_x}}(\cdot, \tau)$ represents the constrained fused-lasso projection operator as defined in [94, 90].

Remark 10. An alternative way to solve the blind deconvolution problems posed in this work is to lift them to a higher-order space and use convex blind deconvolution approaches to arrive at the solutions [91, 100, 92, 97, 101]. In our case, however, one would have to first recover very high-dimensional low-rank tensors and then perform tensor factorization to recover the radar scene and obtain the unknown errors. This approach would be very expensive in terms of memory and computational complexity. Our approach, in contrast, solves a number of convex problems in the original low-dimensional space in a serial fashion to recover the scene. Thus, our proposed method enjoys low memory and computational complexities compared to the existing convex approaches for (multilinear) blind deconvolution.

4.4 Error bounds for blind deconvolution

In this section we derive error bounds for the algorithm prescribed in the previous section, and show that block coordinate descent for the proposed deconvolution problems provides a solution that is very close to the true solution.

4.4.1 BloGD error bounds for generalized model

Since the model proposed in Prop. 2 is the most general model with both ambiguities, we start by deriving error bounds for this model. Afterwards, the error bounds for individual ambiguities can be derived along the same lines as for the generalized model. These error bounds are presented in the following theorem:

Theorem 12. *Consider the distributed radar imaging model with both position and synchronization ambiguities in Prop. 2, and the associated blind deconvolution problem posed in* (4.13). *Then using Alg.3, when* $F = \mathcal{O}(s \log^4 N)$, *the errors of the estimated radar scene and the ambiguities satisfy:*

$$\|\mathbf{h}_m - \mathbf{h}_m^*\|_2^2 \leq \left(\frac{2\|\widetilde{\mathbf{n}}_m\|_2 + 1}{\sigma_{\min \mathbf{C}_\mathbf{x}} \sqrt{F}}\right)^2,$$
$$\|\mathbf{g}_m - \mathbf{g}_m^*\|_2^2 \leq \left(\frac{2\|\widetilde{\mathbf{n}}_m\|_2 + 1}{\sigma_{\min \mathbf{C}_\mathbf{x}} \sqrt{F}}\right)^2, \tag{4.14}$$

and

$$\|\mathbf{x} - \mathbf{x}^*\|_2^2 + \sum_{m=1}^{M} \|\mathbf{z}_m - \mathbf{z}_m^*\|_2^2$$
$$\leq \left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2 + \sqrt{s} + \sqrt{s_{TV}} + 1}{\min\{\widetilde{y}_{\min}, 1\}\sqrt{2FM}}\right)^2, \tag{4.15}$$

by choosing $\beta \leq \frac{\widetilde{y}_{\min}}{M\sqrt{2F}}$ *and* $\alpha \leq \frac{\sigma_{\min \mathbf{C}_\mathbf{x}}}{\sqrt{4F}}$. *Here* s *and* s_{TV} *denote the sparsity of the radar scene with respect to* ℓ_1 *and TV norms, respectively. Moreover,* $\widetilde{y}_{\min}$ *is the magnitude of the smallest entry in all observations and* $\sigma_{\min \mathbf{C}_\mathbf{x}}$ *is the smallest singularvalue of convolution matrix* $\mathbf{C}_\mathbf{x}$ *for two-dimensional convolution with* $\mathbf{x}$.

The proof of this theorem is given in Appendix 4.7.1. The theorem states that for each unknown variable the estimate lies within a ball of certain radius around the true value of the variable. The radius of this ball decreases as the noise in the observations decreases and as the variables become sparser. The radius also decreases as the smallest value in the observations, the bandwidth of the transmitted wave, and the number of observations increase.

Remark 11. The proof in Appendix 4.7.1 also provides error bounds when the parameters are shared among observations, i.e., when one transmitter is shared by multiple receivers and when a receiver receives from multiple transmitters. In this case, the condition on

the required bandwidth gets relaxed to $FM' = \mathcal{O}(s \log^4 N)$, where M' is the number of transmitters/receivers that are shared.

4.4.2 BloGD error bounds for position ambiguity

The error bounds for the case of only position ambiguity are presented in the following algorithm:

Theorem 13. *Consider the distributed radar imaging model with position ambiguities expressed entrywise in Prop. 1, and the associated blind deconvolution problem posed in* (4.11). *Then using Alg.3, when $F = \mathcal{O}(s \log^4 N)$, the errors of the estimated radar scene and the ambiguities satisfy:*

$$\begin{aligned} \|\mathbf{h}_m - \mathbf{h}_m^*\|_2^2 &\leq \left(\frac{2\|\widetilde{\mathbf{n}}_m\|_2 + 1}{\sigma_{\min \mathbf{C_x}} \sqrt{F}} \right)^2, \\ \|\mathbf{g}_m - \mathbf{g}_m^*\|_2^2 &\leq \left(\frac{2\|\widetilde{\mathbf{n}}_m\|_2 + 1}{\sigma_{\min \mathbf{C_x}} \sqrt{F}} \right)^2, \end{aligned} \tag{4.16}$$

and

$$\|\mathbf{x} - \mathbf{x}^*\|_2^2 \leq \left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2 + \sqrt{s} + \sqrt{s_{TV}}}{\sqrt{2FM}} \right)^2, \tag{4.17}$$

by choosing $\alpha \leq \frac{\sigma_{\min \mathbf{C_x}}}{\sqrt{4F}}$. Here s and s_{TV} denote the sparsity of the radar scene with respect to ℓ_1 and TV norms, respectively. Moreover, $\sigma_{\min \mathbf{C_x}}$ is the smallest singularvalue of two-dimensional convolution matrix $\mathbf{C_x}$ for convolution with $\mathbf{x}$.

The proof of this theorem is omitted due to space constraints and follows trivially from the proof of Theorem 12. The error bounds for the position ambiguities in this theorem are similar to the ones in the generalized model, while the bound for the estimated radar scene is slightly different as there is no clock mismatch in this scenario.

4.4.3 BloGD error bounds for clock mismatch

Finally, for the error bounds in the presence of clock mismatch while using Alg. 3 to estimate the unknown, results are presented in the following theorem:

Theorem 14. *Consider the distributed radar imaging model with position ambiguities expressed entrywise in* (4.9)*, and the associated blind deconvolution problem posed in* (4.12)*. Then using Alg.3, when $F = \mathcal{O}(s \log^4 N)$, the error of the estimated radar scene and the clock mismatch satisfies:*

$$\|\mathbf{x} - \mathbf{x}^*\|_2^2 + \sum_{m=1}^{M} \|\mathbf{z}_m - \mathbf{z}_m^*\|_2^2 \leq \left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2 + \sqrt{s} + \sqrt{s_{TV}} + 1}{\min\{\widetilde{y}_{\min}, 1\}\sqrt{2FM}} \right)^2, \tag{4.18}$$

by choosing $\beta \leq \frac{\widetilde{y}_{\min}}{M\sqrt{2F}}$. Here s and s_{TV} denote the sparsity of the radar scene with respect to ℓ_1 and TV norms, respectively. Moreover, $\widetilde{y}_{\min}$ is the magnitude of the smallest entry in all observations.

The proof of this theorem follows closely from the proof of Theorem 12 and is thus omitted.

4.5 Numerical Experiments

We now evaluate the performance of BloGD through numerical experiments on simulated data. For this purpose, we simulated a distributed radar imaging scenario where a scene is imaged with 32 distributed antennas grouped into four arrays of eight antennas each, as shown in Fig. 4.1. The assumed antenna positions (true or erroneous) are always represented as solid dots whereas the actual positions are represented as crosses (as in Fig. 4.2 and Fig. 4.6). In our simulated setup, the transmitting antennas transmit a differential Gaussian pulse of 9 GHz bandwidth centered at 6 GHz. The observations are contaminated with

white Gaussian noise with peak signal to noise ratios (PSNRs) of levels 6 dB, 8 dB, 10 dB, 15 dB, 20 dB, and 25dB, after matched-filtering with the transmitted pulse.

We generate 5 different radar scenes layout each with a different arrangement of three targets within the scene. We also experiment with ten different realizations of the antenna position errors that have an average absolute value of 2λ, where $\lambda = 6$ Ghz is the wavelength of the central frequency of our transmitted pulse. In all experiments with clock mismatch, the clock drift is picked uniformly at random within the range of [-10, 10] time periods. In all cases, we pick $\gamma = 0.5$ and $\alpha = \beta = 7$/PSNR.

For performance comparison, we compare the performance of our proposed approach BloGD that tries to estimate the errors along with the radar scene, with the plain fused Lasso approach that does not assume any errors in the array parameters. Fig. 4.7 shows the results of these experiment for three of the five simulated target scenes at a PSNR of 15 dB. One can clearly see that our proposed models that recover the radar scene as well as the ambiguities outperform the fused Lasso approach by visibly reducing the artifacts that contribute to false alarms, and by detecting all targets in the scene which results in a higher detection rate. A comparison of Receiver operating characteristic (ROC) curves for all three models under various PSNR levels can be seen in Fig. 4.8. The figure highlights the superior performance of our proposed models and the proposed algorithm over the fused Lasso reconstructions with existing models.

4.6 Conclusion

In this paper we developed novel forward models to enable high resolution distributed radar imaging under ambiguous array parameters. We proposed these forward models in the image and time domains, rather than the measurement domain, for exact modeling of position ambiguity in antenna locations and clock mismatch between antennas. We then devised a block coordinate descent based algorithm, called BloGD, for radar scene recovery from the nonconvex multilinear blind deconvlution problems formulated through our proposed for-

ward models. We demonstrated the superior performance of the proposed method through numerical simulations on synthetic data. In the future, we plan to investigate the effects of multipath in our proposed framework and develop precise models for alleviating its effects.

4.7 Appendix

4.7.1 Proof of Theorem 12

The general idea of the proof is similar to the works in [102, 74, 103]. To begin, we first realize that for each observation, the data fidelity term in the objective function can be expressed as:

$$\begin{aligned}\|\widetilde{\mathbf{y}}_m - \mathbf{D}_{\overline{\mathbf{z}}_m}\mathbf{A}_m\mathbf{x}\|_2^2 &= \|\mathbf{D}_{\overline{\mathbf{p}}_m}\widetilde{\mathbf{y}}_m - \mathbf{A}_m\mathbf{x}\|_2^2 \\ &= \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\overline{\mathbf{p}}_m - \mathbf{A}_m\mathbf{x}\|_2^2 \\ &= \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m - \mathbf{A}_m\mathbf{x}\|_2^2, \end{aligned} \tag{4.19}$$

because $\overline{\mathbf{z}}_m$ is a phase only vector, and $\overline{\mathbf{p}}_m = \mathbf{F}\mathbf{p}_m$ is the Fourier transform of the convolutive inverse of $\mathbf{z}_m$ such that $\mathbf{p}_m * \mathbf{z}_m = \delta$. Let us define the difference between a candidate radar scene $\mathbf{x}$ and the true scene $\mathbf{x}^*$, as $\boldsymbol{\Delta}_{\mathbf{x}} = \mathbf{x} - \mathbf{x}^*$. Similarly, define the error between a candidate clock drift $\mathbf{p}_m$ and the true drift $\mathbf{p}_m^*$, as $\boldsymbol{\Delta}_{\mathbf{p}_m} = \mathbf{p}_m - \mathbf{p}_m^*$. Now, define the

following function:

$$
\begin{aligned}
&F_{\mathbf{xp}}(\boldsymbol{\Delta}_{\mathbf{x}}, \boldsymbol{\Delta}_{\mathbf{p}_m}) \\
&= \sum_{m=1}^{M} \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m - \mathbf{A}_m\mathbf{x}\|_2^2 - \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m^* - \mathbf{A}_m\mathbf{x}^*\|_2^2 \\
&= \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \mathbf{p}_m \\ \mathbf{x} \end{bmatrix} \right\|_2^2 - \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \mathbf{p}_m^* \\ \mathbf{x}^* \end{bmatrix} \right\|_2^2 \\
&= \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \mathbf{p}_m^* + \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \right\|_2^2 - \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \mathbf{p}_m^* \\ \mathbf{x}^* \end{bmatrix} \right\|_2^2 \\
&= \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \right\|_2^2 + 2 \begin{bmatrix} \mathbf{p}_m^* \\ \mathbf{x}^* \end{bmatrix}^T \widetilde{\mathbf{A}}_m^H \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \\
&= \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \right\|_2^2 - 2\widetilde{\mathbf{n}}_m^H \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix}
\end{aligned}
\tag{4.20}
$$

where $\widetilde{\mathbf{A}}_m = [\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F},\ -\mathbf{A}_m]$, and $\widetilde{\mathbf{n}}_m = \widetilde{\mathbf{A}}_m \begin{bmatrix} \mathbf{p}_m^* \\ \mathbf{x}^* \end{bmatrix} = \mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m^* - \mathbf{A}_m\mathbf{x}^*$ is the (modified) noise in the m-th observation. Let us also define another function:

$$
\begin{aligned}
F_{\mathbf{x}}(\boldsymbol{\Delta}_{\mathbf{x}}) &= \|\mathbf{x}\|_1 + \gamma\|\mathbf{x}\|_{TV} - (\|\mathbf{x}^*\|_1 + \gamma\|\mathbf{x}^*\|_{TV}) \\
&= \|\mathbf{x}\|_1 - \|\mathbf{x}^*\|_1 + \gamma\|\mathbf{x}\|_{TV} - \gamma\|\mathbf{x}^*\|_{TV} \\
&= \|\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}}\|_1 - \|\mathbf{x}^*\|_1 + \gamma\|\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}}\|_{TV} - \gamma\|\mathbf{x}^*\|_{TV} \\
&= \|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_S\|_1 + \|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_{S^c}\|_1 - \|(\mathbf{x}^*)_S\|_1 - \|(\mathbf{x}^*)_{S^c}\|_1 \\
&\quad + \gamma\|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}}\|_{TV} + \gamma\|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}^c}\|_{TV} \\
&\quad - \gamma\|(\mathbf{x}^*)_{S_{TV}}\|_{TV} - \gamma\|(\mathbf{x}^*)_{S_{TV}^c}\|_{TV} \\
&= \|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_S\|_1 - \|(\mathbf{x}^*)_S\|_1 + \|(\boldsymbol{\Delta}_{\mathbf{x}})_{S^c}\|_1 \\
&+ \gamma\|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}}\|_{TV} - \gamma\|(\mathbf{x}^*)_{S_{TV}}\|_{TV} + \gamma\|(\boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}^c}\|_{TV}
\end{aligned}
\tag{4.21}
$$

where $\|\mathbf{x}\|_{TV} = \|\mathbf{D}_{TV}\mathbf{x}\|_{2,1}$, $\mathbf{D}_{TV} : \mathbb{C}^N \to \mathbb{C}^{2\times N}$ is the two-dimensional finite difference operator, S represents the true support of $\mathbf{x}$, and S_{TV} is the true support of $\mathbf{D}_{TV}\mathbf{x}$. Moreover, $\|(\mathbf{x}^*)_{S^c}\|_1 = 0$, and $\|(\mathbf{x}^*)_{S_{TV}^c}\|_{TV} = 0$ as well. In a similar vein, define

$$
\begin{aligned}
F_{\mathbf{p}}(\boldsymbol{\Delta}_{\mathbf{p}_m}) &= \|\mathbf{p}_m\|_1 - \|\mathbf{p}_m^*\|_1 \\
&= \|(\mathbf{p}_m)_S\|_1 + \|(\mathbf{p}_m)_{S^c}\|_1 - \|(\mathbf{p}_m^*)_S\|_1 - \|(\mathbf{p}_m^*)_{S^c}\|_1 \\
&= \|(\mathbf{p}_m^* + \boldsymbol{\Delta}_{\mathbf{p}_m})_S\|_1 - \|(\mathbf{p}_m^*)_S\|_1 + \|(\boldsymbol{\Delta}_{\mathbf{p}_m})_{S^c}\|_1
\end{aligned}
\tag{4.22}
$$

To derive the bound on the error between the true variables ($\mathbf{x}^*$ and $\mathbf{p}_m^*$) and the estimated variables ($\widehat{\mathbf{x}}$ and $\widehat{\mathbf{p}}_m$), let us first define the following convex function of $\boldsymbol{\Delta}_{\mathbf{x}}$ and

$\mathbf{\Delta}_{\mathbf{p}_m}$:

$$F(\mathbf{\Delta}_{\mathbf{x}}, \mathbf{\Delta}_{\mathbf{p}_m}) = F_{\mathbf{xp}}(\mathbf{\Delta}_{\mathbf{x}}, \mathbf{\Delta}_{\mathbf{p}_m}) \; + \; \mu F_{\mathbf{x}}(\mathbf{\Delta}) + \beta \sum_{m=1}^{M} F_{\mathbf{p}}(\mathbf{\Delta}_{\mathbf{p}_m}), \tag{4.23}$$

and see that this function is minimized at $(\mathbf{\Delta}_{\widehat{\mathbf{x}}}, \mathbf{\Delta}_{\widehat{\mathbf{p}}_m})$ because $\mathbf{\Delta}_{\widehat{\mathbf{x}}}$ and $\mathbf{\Delta}_{\widehat{\mathbf{p}}_m}$ minimize (4.13) for all other unknowns fixed. Also see that for this minima of $F(\mathbf{\Delta}_{\mathbf{x}}, \mathbf{\Delta}_{\mathbf{p}_m})$ we have, $F(\mathbf{\Delta}_{\widehat{\mathbf{x}}}, \mathbf{\Delta}_{\widehat{\mathbf{p}}_m}) \leq F(\mathbf{0}, \mathbf{0}) = 0$.

Our main objective then is to prove that $F(\mathbf{\Delta}_{\mathbf{x}}, \mathbf{\Delta}_{\mathbf{p}_m}) > 0$, for $\mathbf{\Delta}_{\mathbf{x}}$ and $\mathbf{\Delta}_{\mathbf{p}_m}$ of some prescribed norms, respectively. If that can be proved, then since $F(\mathbf{\Delta}_{\widehat{\mathbf{x}}}, \mathbf{\Delta}_{\widehat{\mathbf{p}}_m}) \leq 0$, then $\mathbf{\Delta}_{\mathbf{x}}$ and $\mathbf{\Delta}_{\mathbf{p}_m}$ must have norms smaller than the prescribed norms. To show that

$F(\boldsymbol{\Delta}_{\mathbf{x}}, \boldsymbol{\Delta}_{\mathbf{p}_m}) > 0$, proceed by realizing that:

$$
\begin{aligned}
&F(\boldsymbol{\Delta}_{\mathbf{x}}, \boldsymbol{\Delta}_{\mathbf{p}_m}) = F_{\mathbf{xp}}(\boldsymbol{\Delta}_{\mathbf{x}}, \boldsymbol{\Delta}_{\mathbf{p}_m}) + \mu F_{\mathbf{x}}(\boldsymbol{\Delta}) + \beta \sum_{m=1}^{M} F_{\mathbf{p}}(\boldsymbol{\Delta}_{\mathbf{p}_m}) \\
&= \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \right\|_2^2 - 2\widetilde{\mathbf{n}}_m^H \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \\
&+ \mu \Big(\|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_S\|_1 - \|(\mathbf{x}^*)_S\|_1 + \|(\boldsymbol{\Delta}_{\mathbf{x}})_{S^c}\|_1 \\
&+ \gamma \|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}}\|_{TV} - \gamma \|(\mathbf{x}^*)_{S_{TV}}\|_{TV} + \gamma \|(\boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}^c}\|_{TV} \Big) \\
&+ \beta \Big(\|(\mathbf{p}_m^* + \boldsymbol{\Delta}_{\mathbf{p}_m})_S\|_1 - \|(\mathbf{p}_m^*)^S\|_1 + \|(\boldsymbol{\Delta}_{\mathbf{p}_m})_{S^c}\|_1 \Big) \\
&\overset{(a)}{\geq} \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \right\|_2^2 - 2\widetilde{\mathbf{n}}_m^H \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \\
&+ \mu \Big(\|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_S\|_1 - \|(\mathbf{x}^*)_S\|_1 \\
&+ \gamma \|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}}\|_{TV} - \gamma \|(\mathbf{x}^*)_{S_{TV}}\|_{TV} \Big) \\
&+ \beta \Big(\|(\mathbf{p}_m^* + \boldsymbol{\Delta}_{\mathbf{p}_m})_S\|_1 - \|(\mathbf{p}_m^*)^S\|_1 \Big) \\
&\overset{(b)}{\geq} \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \right\|_2^2 - 2\widetilde{\mathbf{n}}_m^H \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \\
&- \mu \Big| \|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_S\|_1 - \|(\mathbf{x}^*)_S\|_1 \Big| \\
&- \mu\gamma \Big| \|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}}\|_{TV} - \gamma \|(\mathbf{x}^*)_{S_{TV}}\|_{TV} \Big| \\
&- \beta \Big| \|(\mathbf{p}_m^* + \boldsymbol{\Delta}_{\mathbf{p}_m})_S\|_1 - \|(\mathbf{p}_m^*)^S\|_1 \Big| \\
&\overset{(c)}{\geq} \sum_{m=1}^{M} \left\| \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \right\|_2^2 - 2\widetilde{\mathbf{n}}_m^H \widetilde{\mathbf{A}}_m \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_{\mathbf{x}} \end{bmatrix} \\
&- \mu \|(\boldsymbol{\Delta}_{\mathbf{x}})_S\|_1 - \mu\gamma \|(\boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}}\|_{TV} - \beta \|(\boldsymbol{\Delta}_{\mathbf{p}_m})_S\|_1,
\end{aligned} \tag{4.24}
$$

where inequality (a) follows from the fact that $\|(\boldsymbol{\Delta}_{\mathbf{x}})_{S^c}\|_1$, $\|(\boldsymbol{\Delta}_{\mathbf{x}})_{S_{TV}^c}\|_{TV}$, and $\|(\boldsymbol{\Delta}_{\mathbf{p}_m})_{S^c}\|_1$ are all ≥ 0. Additionally, (b) follows from inequalities similar to $\|(\mathbf{x}^* + \boldsymbol{\Delta}_{\mathbf{x}})_S\|_1 -$

$\|(\mathbf{x}^*)_S\|_1 \geq -\Big|\|(\mathbf{x}^* + \boldsymbol{\Delta}_\mathbf{x})_S\|_1 - \|(\mathbf{x}^*)_S\|_1\Big|$, and finally (c) is true because of inequalities similar to $\Big|\|(\mathbf{x}^* + \boldsymbol{\Delta}_\mathbf{x})_S\|_1 - \|(\mathbf{x}^*)_S\|_1\Big| \leq \|(\boldsymbol{\Delta}_\mathbf{x})_S\|_1$.

To further lower bound $F(\boldsymbol{\Delta}_\mathbf{x}, \boldsymbol{\Delta}_{\mathbf{p}_m})$, we will use the inequality $\|(\boldsymbol{\Delta}_\mathbf{x})_S\|_1 \leq \sqrt{s}\|\boldsymbol{\Delta}_\mathbf{x}\|_2$, where s is the number of nonzero entries in $\mathbf{x}$. We will also use the following inequality:

$$\begin{aligned}
\gamma\|(\boldsymbol{\Delta}_\mathbf{x})_{S_{TV}}\|_{TV} &= \gamma\|(\mathbf{D}_{TV}\boldsymbol{\Delta}_\mathbf{x})_{S_{TV}}\|_{2,1} \\
&\leq \gamma\|\mathbf{D}_{TV}\boldsymbol{\Delta}_\mathbf{x}\|_{2,1} \leq \gamma\sqrt{s_{TV}}\|\mathbf{D}_{TV}\boldsymbol{\Delta}_\mathbf{x}\|_2 \\
&\leq \gamma\sqrt{s_{TV}}\sigma_{\max\mathbf{D}}\|\boldsymbol{\Delta}_\mathbf{x}\|_2 \leq \sqrt{s_{TV}}\|\boldsymbol{\Delta}_\mathbf{x}\|_2
\end{aligned} \tag{4.25}$$

by picking $\gamma\sigma_{\max\mathbf{D}} \leq 1$, and where $\sigma_{\max\mathbf{D}}$ is the largest singular value of $\mathbf{D}$. With these we express the lower bound as:

$$\begin{aligned}
F(\boldsymbol{\Delta}_\mathbf{x}, \boldsymbol{\Delta}_{\mathbf{p}_m}) &\geq \sum_{m=1}^{M}\left\|\widetilde{\mathbf{A}}_m \begin{bmatrix}\boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_\mathbf{x}\end{bmatrix}\right\|_2^2 - 2\widetilde{\mathbf{n}}_m^H\widetilde{\mathbf{A}}_m \begin{bmatrix}\boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_\mathbf{x}\end{bmatrix} \\
&- \mu\|(\boldsymbol{\Delta}_\mathbf{x})_S\|_1 - \mu\gamma\|(\boldsymbol{\Delta}_\mathbf{x})_{S_{TV}}\|_{TV} - \beta\|(\boldsymbol{\Delta}_{\mathbf{p}_m})_S\|_1 \\
&\geq \sum_{m=1}^{M}\left\|\widetilde{\mathbf{A}}_m \begin{bmatrix}\boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_\mathbf{x}\end{bmatrix}\right\|_2^2 - 2\widetilde{\mathbf{n}}_m^H\widetilde{\mathbf{A}}_m \begin{bmatrix}\boldsymbol{\Delta}_{\mathbf{p}_m} \\ \boldsymbol{\Delta}_\mathbf{x}\end{bmatrix} \\
&- \mu\sqrt{s}\|\boldsymbol{\Delta}_\mathbf{x}\|_2 - \mu\sqrt{s_{TV}}\|\boldsymbol{\Delta}_\mathbf{x}\|_2 - \beta\sqrt{s_p}\|\boldsymbol{\Delta}_{\mathbf{p}_m}\|_2 \\
&= \|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2^2 - 2\widetilde{\mathbf{n}}^H\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}} - \mu\sqrt{s}\|\boldsymbol{\Delta}_\mathbf{x}\|_2 \\
&\quad - \mu\sqrt{s_{TV}}\|\boldsymbol{\Delta}_\mathbf{x}\|_2 - \beta\sum_{m=1}^{M}\sqrt{s_p}\|\boldsymbol{\Delta}_{\mathbf{p}_m}\|_2 \\
&= \|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2\Bigg[\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2 - 2\frac{\widetilde{\mathbf{n}}^H\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}}{\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\
&\quad - \mu\sqrt{s}\frac{\|\boldsymbol{\Delta}_\mathbf{x}\|_2}{\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} - \mu\sqrt{s_{TV}}\frac{\|\boldsymbol{\Delta}_\mathbf{x}\|_2}{\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\
&\quad - \beta\sum_{m=1}^{M}\sqrt{s_p}\frac{\|\boldsymbol{\Delta}_{\mathbf{p}_m}\|_2}{\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2}\Bigg],
\end{aligned} \tag{4.26}$$

where the vector $\widetilde{\mathbf{n}} = [\widetilde{\mathbf{n}}_1^T, \widetilde{\mathbf{n}}_2^T, \cdots, \widetilde{\mathbf{n}}_M^T]^T$, the vector $\boldsymbol{\Delta}_{\mathbf{xp}} = [\boldsymbol{\Delta}_\mathbf{x}^T, \boldsymbol{\Delta}_{\mathbf{p}_1}^T, \boldsymbol{\Delta}_{\mathbf{p}_2}^T, \cdots, \boldsymbol{\Delta}_{\mathbf{p}_M}^T]^T$,

and the matrix

$$\mathbf{A_{xp}} = \begin{bmatrix} \mathbf{A}_1 & -\mathbf{D}_{\widetilde{\mathbf{y}}_1}\mathbf{F} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{0} & -\mathbf{D}_{\widetilde{\mathbf{y}}_2}\mathbf{F} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \ddots & \mathbf{0} \\ \mathbf{A}_M & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{D}_{\widetilde{\mathbf{y}}_M}\mathbf{F} \end{bmatrix}. \tag{4.27}$$

In the following we will use several facts, such as: (i) cosine of the angle between two F-dimensional random Gaussian vectors with real entries is upper-bounded by $1/\sqrt{F}$ [104], and (ii) with complex entries is bounded by $1/\sqrt{2F}$, and that (iii) $\sqrt{s_p} = 1$.Using these facts, the inequality in (4.26) will still be true, if the following slightly different inequality is satisfied:

$$\begin{aligned} \frac{F(\boldsymbol{\Delta}_{\mathbf{x}}, \boldsymbol{\Delta}_{\mathbf{p}_m})}{\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} &\geq \|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2 - 2\frac{\widetilde{\mathbf{n}}^H\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}}{\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\ &\quad - \mu\sqrt{s}\frac{\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2}{\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} - \mu\sqrt{s_{TV}}\frac{\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2}{\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\ &\quad - \beta\sum_{m=1}^{M}\sqrt{s_p}\frac{\|\boldsymbol{\Delta}_{\mathbf{p}_m}\|_2}{\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\ &\geq \|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2 - 2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}} - \mu(\sqrt{s} + \sqrt{s_{TV}})\frac{\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2}{\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\ &\quad - \frac{\beta\sum_{m=1}^{M}\|\boldsymbol{\Delta}_{\mathbf{p}_m}\|_2}{\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2}. \end{aligned} \tag{4.28}$$

Let us briefly focus on $\|\mathbf{A_{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2$. In matrix $\mathbf{A_{xp}}$, each sub-matrix $\mathbf{A}_m$ is similar to a Fourier matrix (both have similar coherence properties) [105], and the coherence decreases with increasing dimensions of the radar scene (increasing number of rows in each sub-matrix $\mathbf{A}_m$) and increasing distance between the scene and the radar arrays. Therefore,

with the results in [105] while considering the columns to be normalized:

$$
\begin{aligned}
&\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2^2 \\
&= \left\| \begin{bmatrix} \mathbf{A}_1 & -\mathbf{D}_{\widetilde{\mathbf{y}}_1}\mathbf{F} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{0} & -\mathbf{D}_{\widetilde{\mathbf{y}}_2}\mathbf{F} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \ddots & \mathbf{0} \\ \mathbf{A}_M & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{D}_{\widetilde{\mathbf{y}}_M}\mathbf{F} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{x}} \\ \boldsymbol{\Delta}_{\mathbf{p}_1} \\ \boldsymbol{\Delta}_{\mathbf{p}_2} \\ \vdots \\ \boldsymbol{\Delta}_{\mathbf{p}_M} \end{bmatrix} \right\|_2^2 \\
&= \left\| \begin{bmatrix} \mathbf{A}_1 & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{0} & -\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \ddots & \mathbf{0} \\ \mathbf{A}_M & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{x}} \\ \widetilde{\mathbf{y}}_1 \odot \mathbf{F}\boldsymbol{\Delta}_{\mathbf{p}_1} \\ \widetilde{\mathbf{y}}_2 \odot \mathbf{F}\boldsymbol{\Delta}_{\mathbf{p}_2} \\ \vdots \\ \widetilde{\mathbf{y}}_M \odot \mathbf{F}\boldsymbol{\Delta}_{\mathbf{p}_M} \end{bmatrix} \right\|_2^2 \\
&\overset{(d)}{\geq} 0.5 \left\| \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{x}} \\ \widetilde{\mathbf{y}}_1 \odot \mathbf{F}\boldsymbol{\Delta}_{\mathbf{p}_1} \\ \widetilde{\mathbf{y}}_2 \odot \mathbf{F}\boldsymbol{\Delta}_{\mathbf{p}_2} \\ \vdots \\ \widetilde{\mathbf{y}}_M \odot \mathbf{F}\boldsymbol{\Delta}_{\mathbf{p}_M} \end{bmatrix} \right\|_2^2 \geq 0.5(\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2 + \widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2)
\end{aligned}
\tag{4.29}
$$

where $\widetilde{y}_{\min}$ is the smallest entry over all observations, and (d) follows with high probability

as long as $FM = \mathrm{O}(s \log^4 N)$ [105]. Now, (4.28) can be further lower-bounded as:

$$
\begin{aligned}
&\frac{F(\boldsymbol{\Delta}_{\mathbf{x}}, \boldsymbol{\Delta}_{\mathbf{p}_m})}{\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\
&\geq \|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2 - 2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}} \\
&\quad - \mu(\sqrt{s} + \sqrt{s_{TV}})\frac{\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2}{\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} - \frac{\beta \sum\limits_{m=1}^{M} \|\boldsymbol{\Delta}_{\mathbf{p}_m}\|_2}{\|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2} \\
&\geq \|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2 - 2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}} \\
&\quad - \mu(\sqrt{s} + \sqrt{s_{TV}})\frac{\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2}{\sqrt{0.5}\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2} - \frac{\beta \sum\limits_{m=1}^{M} \|\boldsymbol{\Delta}_{\mathbf{p}_m}\|_2}{\sqrt{0.5}\widetilde{y}_{\min}\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2} \\
&\geq \|\mathbf{A}_{\mathbf{xp}}\boldsymbol{\Delta}_{\mathbf{xp}}\|_2 - 2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}} - \frac{\mu(\sqrt{s} + \sqrt{s_{TV}})}{\sqrt{0.5}} - \frac{\beta\sqrt{M}}{\sqrt{0.5}\widetilde{y}_{\min}} \\
&\geq \sqrt{0.5(\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2 + \widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2)} - 2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}} \\
&\qquad - \frac{\mu(\sqrt{s} + \sqrt{s_{TV}})}{\sqrt{0.5}} - \frac{\beta\sqrt{M}}{\sqrt{0.5}\widetilde{y}_{\min}}
\end{aligned}
\tag{4.30}
$$

where we used the fact that $\frac{\sum_{m=1}^{M} \|\boldsymbol{\Delta}_{\mathbf{p}m}\|_2}{\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2} = \frac{\sum_{m=1}^{M} \|\boldsymbol{\Delta}_{\mathbf{p}m}\|_2}{\sqrt{\sum_{m=1}^{M} \|\boldsymbol{\Delta}_{\mathbf{p}m}\|_2^2}} \leq \sqrt{M}$. All that's left is to find the condition under which this fina lower bound is greater than zero. To that end, proceed as

follows:

$$
\begin{aligned}
&\sqrt{0.5(\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2+\widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2)}-2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}}\\
&\qquad-\frac{\mu(\sqrt{s}+\sqrt{s_{TV}})}{\sqrt{0.5}}-\frac{\beta\sqrt{M}}{\sqrt{0.5}\widetilde{y}_{\min}}>0\\
\Rightarrow&\sqrt{0.5(\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2+\widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2)}>2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}}\\
&\qquad+\frac{\mu(\sqrt{s}+\sqrt{s_{TV}})}{\sqrt{0.5}}+\frac{\beta\sqrt{M}}{\sqrt{0.5}\widetilde{y}_{\min}}\\
\Rightarrow&\sqrt{0.5(\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2+\widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2)}>2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}}\\
&\qquad+\frac{\mu(\sqrt{s}+\sqrt{s_{TV}})}{\sqrt{0.5}}+\frac{\beta\sqrt{M}}{\sqrt{0.5}\widetilde{y}_{\min}}\\
\Rightarrow&\sqrt{0.5(\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2+\widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2)}>2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2FM}}\\
&\qquad+\frac{\mu(\sqrt{s}+\sqrt{s_{TV}})}{\sqrt{0.5}}+\frac{\beta\sqrt{M}}{\sqrt{0.5}\widetilde{y}_{\min}}\\
\Rightarrow&\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2+\widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2>\left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2+\sqrt{s}+\sqrt{s_{TV}}+1}{\sqrt{2FM}}\right)^2,\\
\Rightarrow&\frac{\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2}{\widetilde{y}_{\min}^2}+\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2>\left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2+\sqrt{s}+\sqrt{s_{TV}}+1}{\widetilde{y}_{\min}\sqrt{2FM}}\right)^2,
\end{aligned}
\tag{4.31}
$$

where we β and μ are chosen such that $\beta \leq \frac{\widetilde{y}_{\min}}{M\sqrt{2F}}$ and $\mu \leq \frac{1}{\sqrt{2FM}}$, respectively. Now, if $\widetilde{y}_{\min} \leq 1$, then $\frac{\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2}{\widetilde{y}_{\min}^2} \geq \|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2$, and the bound can be expressed as:

$$
\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2+\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2>\left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2+\sqrt{s}+\sqrt{s_{TV}}+1}{\widetilde{y}_{\min}\sqrt{2FM}}\right)^2,
\tag{4.32}
$$

otherwise, if $\widetilde{y}_{\min} > 1$, then $\widetilde{y}_{\min}^2\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2 > \|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2$, and the bound can be expressed as:

$$
\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2+\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2>\left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2+\sqrt{s}+\sqrt{s_{TV}}+1}{\sqrt{2FM}}\right)^2.
\tag{4.33}
$$

Combining both cases, the bound can be expressed as:

$$\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2 + \|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2 > \left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2 + \sqrt{s} + \sqrt{s_{TV}} + 1}{\min\{\widetilde{y}_{\min}, 1\}\sqrt{2FM}} \right)^2, \tag{4.34}$$

This implies that for $F(\boldsymbol{\Delta}_{\mathbf{x}}, \boldsymbol{\Delta}_{\mathbf{p}_m})$, the error satisfies:

$$\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2 + \|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2 \leq \left(\frac{\sqrt{2}\|\widetilde{\mathbf{n}}\|_2 + \sqrt{s} + \sqrt{s_{TV}} + 1}{\min\{\widetilde{y}_{\min}, 1\}\sqrt{2FM}} \right)^2, \tag{4.35}$$

which is the required result for error bound on $\|\boldsymbol{\Delta}_{\mathbf{x}}\|_2^2$ and $\|\boldsymbol{\Delta}_{\mathbf{p}}\|_2^2$.

We now shift our attention towards $\mathbf{h}_m$, and we begin by realizing that the data fidelity term in objective function can be expressed to focus on $\mathbf{h}_m$ as:

$$\begin{aligned}
&\|\widetilde{\mathbf{y}}_m - \mathbf{D}_{\bar{\mathbf{z}}_m}\mathbf{A}_m\mathbf{x}\|_2^2 \\
&= \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m - \mathbf{A}_m\mathbf{x}\|_2^2 \\
&= \sum_{w=1}^{F} \|\widetilde{\mathbf{y}}_m(w)[\mathbf{F}\mathbf{p}_m](w) - \mathbf{A}'_w\Big(\big\{(\mathbf{A}_w * \mathbf{h}_m)\,\mathbf{x}\big\} * \mathbf{g}_m\Big)\|_2^2 \\
&= \sum_{w=1}^{F} \|\widetilde{\mathbf{y}}_m(w)[\mathbf{F}\mathbf{p}_m](w) - \mathbf{A}'_w\Big(\big\{\mathbf{A}_w(\mathbf{x} * \mathbf{q}_m)\big\} * \mathbf{h}_m * \mathbf{g}_m\Big)\|_2^2 \\
&= \sum_{w=1}^{F} \|\widetilde{\mathbf{y}}_m(w)[\mathbf{F}\mathbf{p}_m](w) - \mathbf{A}'_w\Big(\big\{\mathbf{A}_w(\mathbf{x} * \mathbf{q}_m)\big\} * \mathbf{t}_m\Big)\|_2^2 \\
&= \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m - \mathbf{B}_m(\mathbf{x} * \mathbf{q}_m)\|_2^2
\end{aligned} \tag{4.36}$$

where $\mathbf{q}_m * \mathbf{h}_m = \delta$, $\mathbf{h}_m * \mathbf{g}_m = \mathbf{t}_m$, and $\mathbf{B}_m$ is the matrix that represents the forward model with focus on $\mathbf{q}_m$. We will make $\mathbf{q}_m$ our subject in order to bound error on $\mathbf{h}_m$. To this end, define $\boldsymbol{\Delta}_{\mathbf{q}_m} = \mathbf{q}_m - \mathbf{q}_m^*$, which is the difference between a candidate shift $\mathbf{q}_m$ and the true

shift $\mathbf{q}_m^*$, and also define the following function:

$$
\begin{aligned}
&F_{\mathbf{q}}(\boldsymbol{\Delta}_{\mathbf{q}_m}) \\
&= \sum_{m=1}^{M} \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m - \mathbf{B}_m(\mathbf{x} * \mathbf{q}_m)\|_2^2 + \alpha\|\mathbf{q}_m\|_1 \\
&\qquad - \|\mathbf{D}_{\widetilde{\mathbf{y}}_m}\mathbf{F}\mathbf{p}_m - \mathbf{B}_m(\mathbf{x} * \mathbf{q}_m^*)\|_2^2 - \alpha\|\mathbf{q}_m^*\|_1 \\
&= \sum_{m=1}^{M} \|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2^2 - 2\widetilde{\mathbf{n}}_m^H\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m}) + \\
&\alpha(\|(\mathbf{q}_m^* + \boldsymbol{\Delta}_{\mathbf{q}_m})_S\|_1 - \|(\mathbf{q}_m^*)_S\|_1 + \|(\boldsymbol{\Delta}_{\mathbf{q}_m})_{S^c}\|_1)
\end{aligned} \tag{4.37}
$$

similar to (4.20) and (4.22). Following along the steps in (4.30), we can lower-bound this as:

$$
\begin{aligned}
&F_{\mathbf{q}}(\boldsymbol{\Delta}_{\mathbf{q}_m}) \\
&= \sum_{m=1}^{M} \|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2^2 - 2\widetilde{\mathbf{n}}_m^H\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m}) + \\
&\alpha(\|(\mathbf{q}_m^* + \boldsymbol{\Delta}_{\mathbf{q}_m})_S\|_1 - \|(\mathbf{q}_m^*)_S\|_1 + \|(\boldsymbol{\Delta}_{\mathbf{q}_m})_{S^c}\|_1) \\
&\geq \sum_{m=1}^{M} \|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2 \Big(\|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2 - 2\frac{\|\widetilde{\mathbf{n}}\|_2}{\sqrt{2F}} \\
&- \frac{\alpha\|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2}{\|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2}\Big).
\end{aligned} \tag{4.38}
$$

We can now lower bound $\|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2^2$ as:

$$
\begin{aligned}
\|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2^2 &\geq 0.5\,\|\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m}\|_2^2 \\
&= 0.5\,\|\mathbf{C}_{\mathbf{x}}\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2^2 \\
&\geq 0.5\sigma_{\min \mathbf{C}_{\mathbf{x}}}^2\,\|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2^2\,,
\end{aligned} \tag{4.39}
$$

if the number of frequency components is $F = \mathrm{O}(s\log^4 N)$ [105], and where $\mathbf{C}_{\mathbf{x}}$ is the

two-dimensional convolution matrix for convolution with $\mathbf{x}$, and $\sigma_{\min \mathbf{C_x}}$ is the minimum singularvalue of this convolution matrix. With this lower bound, we can further lower bound our function from (4.38) as:

$$
\begin{aligned}
&F_{\mathbf{q}}(\boldsymbol{\Delta}_{\mathbf{q}_m}) \\
&\geq \sum_{m=1}^{M} \|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2 \Big(\|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2 - 2\frac{\|\widetilde{\mathbf{n}}_m\|_2}{\sqrt{2F}} \\
&- \frac{\alpha \|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2}{\|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2} \Big) \\
&\geq \sum_{m=1}^{M} \|\mathbf{B}_m(\mathbf{x} * \boldsymbol{\Delta}_{\mathbf{q}_m})\|_2 \Big(\sqrt{0.5}\sigma_{\min \mathbf{C_x}} \|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2 - 2\frac{\|\widetilde{\mathbf{n}}_m\|_2}{\sqrt{2F}} \\
&- \frac{\alpha \|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2}{\sqrt{0.5}\sigma_{\min \mathbf{C_x}} \|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2} \Big),
\end{aligned}
\tag{4.40}
$$

which requires the following to be satisfied for $F_{\mathbf{q}}(\boldsymbol{\Delta}_{\mathbf{q}_m})$ to be greater than zero:

$$
\begin{aligned}
&\sqrt{0.5}\sigma_{\min \mathbf{C_x}} \|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2 - \frac{2\|\widetilde{\mathbf{n}}_m\|_2}{\sqrt{2F}} - \frac{\alpha}{\sqrt{0.5}\sigma_{\min \mathbf{C_x}}} > 0 \\
\Rightarrow &\sqrt{0.5}\sigma_{\min \mathbf{C_x}} \|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2 > \frac{2\|\widetilde{\mathbf{n}}_m\|_2}{\sqrt{2F}} + \frac{\alpha}{\sqrt{0.5}\sigma_{\min \mathbf{C_x}}} \\
\Rightarrow &\|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2 > \frac{2\|\widetilde{\mathbf{n}}_m\|_2}{\sigma_{\min \mathbf{C_x}}\sqrt{F}} + \frac{\alpha}{\sqrt{0.25}\sigma^2_{\min \mathbf{C_x}}} \\
\Rightarrow &\|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2 > \frac{2\|\widetilde{\mathbf{n}}_m\|_2 + 1}{\sigma_{\min \mathbf{C_x}}\sqrt{F}},
\end{aligned}
\tag{4.41}
$$

by choosing $\frac{\alpha\sqrt{F}}{\sqrt{0.25}\sigma_{\min \mathbf{C_x}}} \leq 1 \Rightarrow \alpha \leq \frac{\sigma_{\min \mathbf{C_x}}}{\sqrt{4F}}$. Therefore, the final error bound for $\|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2$ comes out to be:

$$
\|\boldsymbol{\Delta}_{\mathbf{q}_m}\|_2 \leq \frac{2\|\widetilde{\mathbf{n}}_m\|_2 + 1}{\sigma_{\min \mathbf{C_x}}\sqrt{F}}. \tag{4.42}
$$

A fairly common scenario in distributed imaging is the sharing for one transmitter over multiple receivers. If that is the case, and the signal transmitted by the m-th transmitter is received by M' receivers, then another bound can be derived along the steps in (4.30) and

(4.31), given by:

$$\|\mathbf{\Delta}_{\mathbf{q}_m}\|_2 \leq \frac{2\|\widetilde{\mathbf{n}}\|_2 + 1}{\sigma_{\min \mathbf{C}_\mathbf{x}} \sqrt{FM'}}. \tag{4.43}$$

by choosing $\alpha \leq \frac{\sigma_{\min \mathbf{C}_\mathbf{x}}}{\sqrt{4FM'}}$ and the relaxed condition on number of frequency components as $FM' = \mathrm{O}(s \log^4 N)$. Following similar arguments, the error for the other shift kernel can also be derived to be:

$$\|\mathbf{\Delta}_{\mathbf{g}_m}\|_2 \leq \frac{2\|\widetilde{\mathbf{n}}_m\|_2 + 1}{\sigma_{\min \mathbf{C}_\mathbf{x}} \sqrt{F}}, \tag{4.44}$$

when each receiver is receiving from one antenna, and:

$$\|\mathbf{\Delta}_{\mathbf{g}_m}\|_2 \leq \frac{2\|\widetilde{\mathbf{n}}\|_2 + 1}{\sigma_{\min \mathbf{C}_\mathbf{x}} \sqrt{FM'}}, \tag{4.45}$$

when the m-th receiver is receiving from M' different antennas, respectively. ■

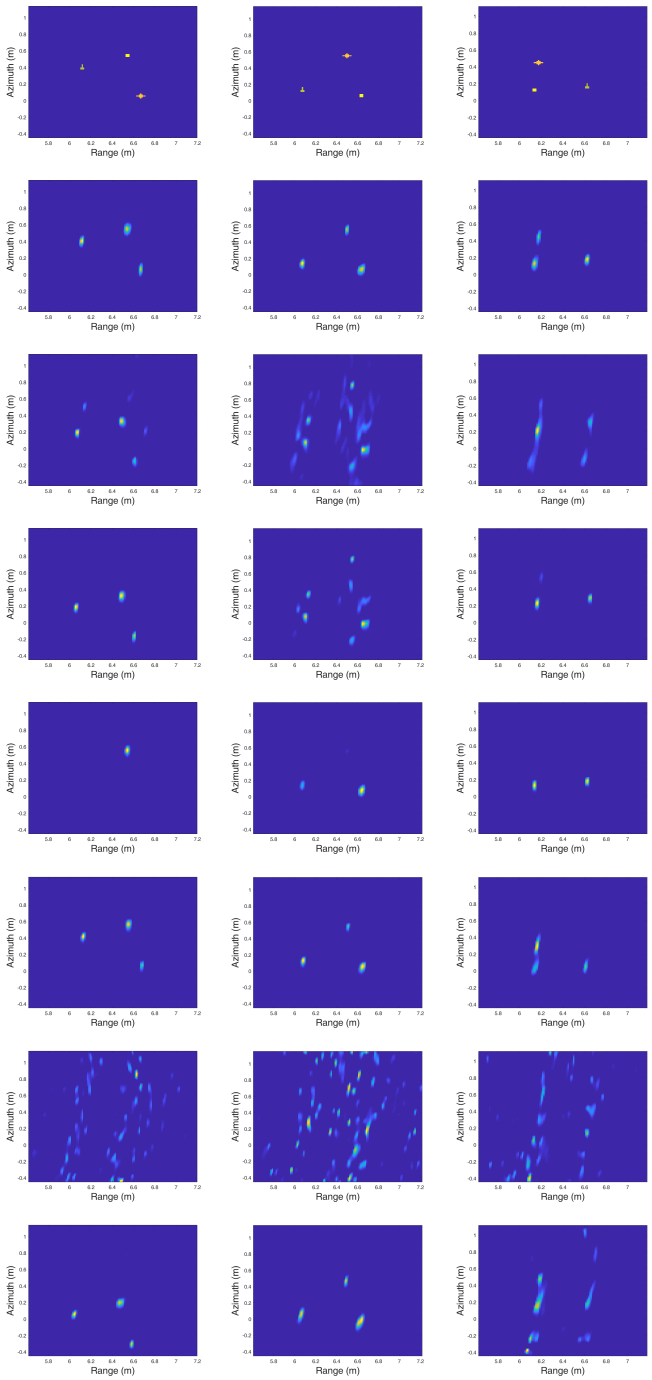

Figure 4.7: This figure shows the results of numerical experiments with three different radar scenes reconstructed at an SNR of 15 dB. Each column represents a different scene, and the rows represent the following in order: ground truth radar scene, fused Lasso reconstruction without any ambiguities, fused Lasso reconstruction suffering clock mismatch, BloGD reconstruction with clock mismatch correction, fused Lasso reconstruction suffering position errors, BloGD reconstruction with position error correction, fused Lasso reconstruction suffering both position errors and clock mismatch, BloGD reconstruction with correction for both ambiguities.

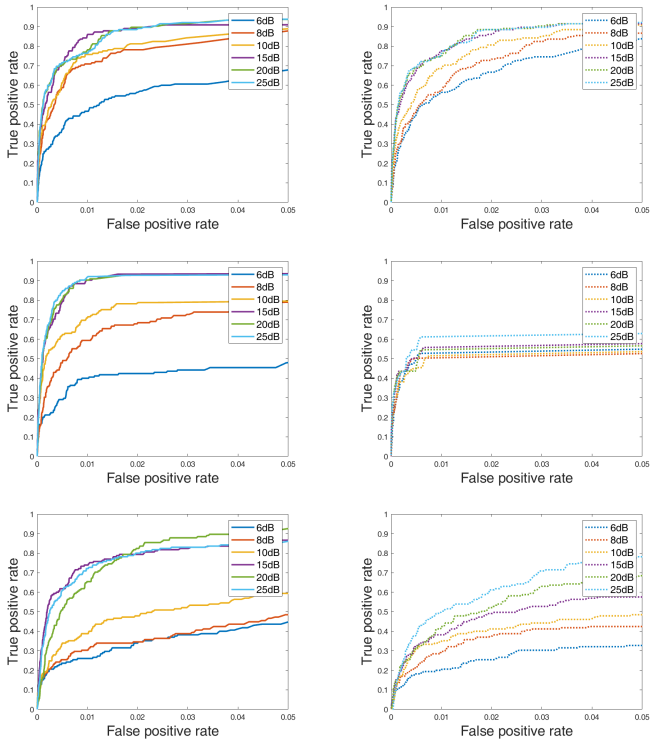

Figure 4.8: This figure compares the ROC curves for all experiments performed under various level of ambiguities and reconstruction with BloGD and fused Lasso. The first row shows the ROC curves for different SNR levels for reconstruction under only clock mismatch with BloGD (left) and fused Lasso (right). The second row shows the curves for reconstruction under position errors, and finally the third rows showcases the results for when both ambiguities are present in the setup. One can see that BloGD outperforms fused Lasso except for all case, except for 6 dB curve for clock mismatch which can be attributed to limited noise realizations in our experiments.

CHAPTER 5
CONCLUSION

To summarize, the objective of this dissertation was to develop methods that exploit structure in various aspects of modern data. To this end, we focused on three specific applications of structure exploitation. In Chapter 2 we focused on detecting and classifying signals under the UoS model. We posed the problems as hypothesis testing problems, derived bounds on performance measures, and expressed the bounds in terms of the geometry between the subspaces and the geometry of the colored noise.

Chapter 3 focused on developing methods to learn unstructured and structured graphs from data through smooth graph signals. We posed the unstructured graph learning problem as a linear program and numerically validated its superior performance over existing state of the art methods. For structured graphs, we made product graphs the focus of our attention, developed an algorithm for learning product graphs in terms of the factor graphs, and derived error bounds for the proposed algorithm. We also validated the performance of the proposed product graph learning algorithm on synthetic and real-world datasets.

Finally, in Chapter 4 we developed precise models for distributed radar imaging with ambiguous array parameters, i.e., imprecise antennas locations and unsynchronized clocks between antennas, using the known properties of the ambiguities. We then posed the radar scene reconstruction problems under said ambiguities as blind deconvolution problems using these known properties and the prior knowledge about the radar scene. Furthermore, we proposed an block coordinate descent based algorithm for solving these problems, derived the associated error bounds, and validated the performance through numerical simulations.

Through our work in this dissertation, we have highlighted the importance of and the advantages gained by successfully exploiting structure in information processing applications. We showcased scenarios where the structure can be exploited in the data itself, the

data acquisition process, or the underlying process generating the data. Our future work will continue these efforts in the problems explored in this dissertation and other related problems.

REFERENCES

[1] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process*, vol. 4, no. 2, pp. 445–460, 2010.

[2] H. L. Yap and R. Pribić, "False alarms in multi-target radar detection within a sparsity framework," in *Proc. IEEE Intl. Radar Conf.*, 2014, pp. 1–6.

[3] M. Joneidi, P. Ahmadi, M. Sadeghi, and N. Rahnavard, "Union of low-rank subspaces detector," *IET Signal Processing*, vol. 10, no. 1, pp. 55–62, 2016.

[4] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[5] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Sig. Proc.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[6] A. Sandryhaila and J. M. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[7] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[8] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[9] L. S. Louis, "Statistical signal processing: Detection, estimation, and time series analysis," *Addision-Wesley Publishing Company*, 1991.

[10] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Upper Saddle River, NJ: Prentice Hall, 1998.

[11] Y. M. Lu and M. N. Do, "A theory for sampling signals from a union of subspaces," *IEEE Trans. Sig. Proc.*, vol. 56, no. 6, pp. 2334–2345, 2008.

[12] T. Wu and W. U. Bajwa, "Revisiting robustness of the union-of-subspaces model for data-adaptive learning of nonlinear signal models," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing (ICASSP'14)*, Florence, Italy, May 2014, pp. 3390–3394.

[13] ——, "Metric-constrained kernel union of subspaces," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing (ICASSP'15)*, Brisbane, Australia, Apr. 2015, pp. 5778–5782.

[14] T. Wu and W. U. Bajwa, "Learning the nonlinear geometry of high-dimensional data: Models and algorithms," *IEEE Trans. Signal Processing*, vol. 63, no. 23, pp. 6229–6244, Dec. 2015.

[15] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," in *International Conference on Learning Representations*, 2019.

[16] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.

[17] R. A. Horn, C. R. Johnson, and L Elsner, *Topics in matrix analysis.* 1994.

[18] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[19] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *IEEE Computer Vision and Pattern Recognition*, IEEE, vol. 1, 2003, pp. 1–11.

[20] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Trans. Image Proc.*, vol. 15, no. 12, pp. 3655–3671, 2006.

[21] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[22] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[23] W. U. Bajwa and D. G. Mixon, "A multiple hypothesis testing approach to low-complexity subspace unmixing," *arXiv preprint arXiv:1408.1469*, 2014.

[24] F. Gini, M Greco, and F Farina, "Radar detection and preclassification based on multiple hypothesis," *IEEE Trans. Aero. Elec. Sys.*, vol. 40, no. 3, pp. 1046–1059, 2004.

[25] T. Wu, P. Gurram, R. M. Rao, and W. U. Bajwa, "Hierarchical union-of-subspaces model for human activity summarization," in *Proc. IEEE Intl. Conf. Comp. Vision*, 2015, pp. 1–9.

[26] T. Wimalajeewa, Y. C. Eldar, and P. K. Varshney, "Subspace recovery from structured union of subspaces," *IEEE Trans. Info. Th.*, vol. 61, no. 4, pp. 2101–2114, 2015.

[27] L. L. Scharf and B. Friedlander, "Matched subspace detectors," *IEEE Trans. Sig. Proc.*, vol. 42, no. 8, pp. 2146–2157, 1994.

[28] S. Kraut, L. L. Scharf, and L. T. McWhorter, "Adaptive subspace detectors," *IEEE Trans. Sig. Proc.*, vol. 49, no. 1, pp. 1–16, 2001.

[29] S. Kraut and L. L. Scharf, "The CFAR adaptive subspace detector is a scale-invariant GLRT," *IEEE Trans. Sig. Proc.*, vol. 47, no. 9, pp. 2538–2541, 1999.

[30] E. J. Kelly, "An adaptive detection algorithm," *IEEE Trans. Aero. Elec. Sys.*, no. 2, pp. 115–127, 1986.

[31] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[32] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc.*, pp. 267–288, 1996.

[33] S. N. Afriat, "Orthogonal and oblique projectors and the characteristics of pairs of vector spaces," in *Mathematical Proceedings of the Cambridge Philosophical Society*, 1957, pp. 800–816.

[34] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis, *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote sensing of environment*, vol. 65, no. 3, pp. 227–248, 1998.

[35] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Patt. Anal. Mach. Intllg.*, vol. 25, no. 2, pp. 218–233, 2003.

[36] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Patt. Anal. Mach. Intllg.*, vol. 23, no. 6, pp. 643–660, 2001.

[37] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *IEEE Comp. Vision and Pattern Recognition*, IEEE, 2007, pp. 1–8.

[38] C. Khatri, P. Krishnaiah, and P. K. Sen, "A note on the joint distribution of correlated quadratic forms," *J. Stat. Planning Inference*, vol. 1, no. 3, pp. 299–307, 1977.

[39] D. Jensen, "The joint distribution of quadratic forms and related distributions1," *Australian J. Statistics*, vol. 12, no. 1, pp. 13–22, 1970.

[40] D. Jensen and H. Solomon, "Approximations to joint distributions of definite quadratic forms," *J. Amer. Statistical Assoc.*, vol. 89, no. 426, pp. 480–486, 1994.

[41] T. Y. Al-Naffouri, M. Moinuddin, N. Ajeeb, B. Hassibi, and A. L. Moustakas, "On the distribution of indefinite quadratic forms in Gaussian random variables," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 153–165, 2016.

[42] D De Caen, "A lower bound on the probability of a union," *Discrete Mathematics*, vol. 169, no. 1-3, pp. 217–220, 1997.

[43] M. Fréchet, "Généralisation du théoreme des probabilités totales," *Fundamenta Mathematicae*, vol. 1, no. 25, pp. 379–387, 1935.

[44] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

[45] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from filtered signals: Graph system and diffusion kernel identification," *IEEE Transactions on Signal and Information Processing over Networks*, 2018.

[46] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 467–483, 2017.

[47] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing*, 2017, pp. 6508–6512.

[48] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, 2017.

[49] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning laplacian matrix in smooth graph signal representations,” *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[50] V. N. Ioannidis, Y. Shen, and G. B. Giannakis, “Semi-blind inference of topologies and dynamical processes over graphs,” *arXiv preprint arXiv:1805.06095*, 2018.

[51] V. Kalofolias, “How to learn a graph from smooth signals,” in *Proc. Artificial Intelligence and Statistics*, 2016, pp. 920–929.

[52] S. Chen, R. Varma, A. Singh, and J. Kovačević, “Signal recovery on graphs: Fundamental limits of sampling strategies,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 539–554, 2016.

[53] E. Ceci, Y. Shen, G. B. Giannakis, and S. Barbarossa, “Signal and graph perturbations via total least-squares,” in *Proc. Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2018, pp. 747–751.

[54] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, “Graph topology inference based on sparsifying transform learning,” *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1712–1727, 2019.

[55] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[56] K. Greenewald, S. Zhou, and A. Hero III, “Tensor graphical lasso (teralasso),” *arXiv preprint arXiv:1705.03983*, 2017.

[57] T. Tsiligkaridis and A. O. Hero, “Covariance estimation in high dimensions via kronecker product expansions,” *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5347–5360, 2013.

[58] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[59] S. Wang and N. Shroff, “A new alternating direction method for linear programming,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1480–1488.

[60] J. Eckstein, D. P. Bertsekas, *et al.*, “An alternating direction method for linear programming,” 1990.

[61] G. I. Allen and R. Tibshirani, “Transposable regularized covariance models with an application to missing data imputation,” *The Annals of Applied Statistics*, vol. 4, no. 2, p. 764, 2010.

[62] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 985–1042, 2010.

[63] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014. arXiv: `1408.5781 [cs.IT]`.

[64] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, *et al.*, "The ncep/ncar 40-year reanalysis project," *Bulletin of the American Meteorological Society*, vol. 77, no. 3, pp. 437–472, 1996.

[65] D. G. Luenberger, *Optimization by Vector Space Methods*. John Wiley & Sons, 1997.

[66] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning based on sparseness of temporal variation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 5411–5415.

[67] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Ieee, 2017, pp. 2826–2830.

[68] C. Craddock, Y. Benhajali, C. Chu, F. Chouinard, A. Evans, A. Jakab, B. S. Khundrakpam, J. D. Lewis, Q. Li, M. Milham, *et al.*, "The Neuro Bureau Preprocessing Initiative: Open sharing of preprocessed neuroimaging data and derivatives," *Neuroinformatics*, 2013.

[69] M. Narayan, *Preprocessed abide dataset (UM and UCLA)*, 2015.

[70] M. Narayan and G. I. Allen, "Mixed effects models for resampled network statistics improves statistical power to find differences in multi-subject functional connectivity," *Frontiers in Neuroscience*, vol. 10, p. 108, 2016.

[71] A. Dobra, "Variable selection and dependency networks for genomewide data," *Biostatistics*, vol. 10, no. 4, pp. 621–639, 2009.

[72] L. Li and K.-C. Toh, "An inexact interior point method for l 1-regularized sparse covariance selection," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 291–315, 2010.

[73] Y. Cui, C. Leng, and D. Sun, "Sparse estimation of high-dimensional correlation matrices," *Computational Statistics & Data Analysis*, vol. 93, pp. 390–403, 2016.

[74] W. Sun, Z. Wang, H. Liu, and G. Cheng, "Non-convex statistical optimization for sparse tensor graphical model," in *Advances in neural information processing systems*, 2015, pp. 1081–1089.

[75] Y. Fang, K. A. Loparo, and X. Feng, "Inequalities for the trace of matrix product," *IEEE Transactions on Automatic Control*, vol. 39, no. 12, pp. 2489–2490, 1994.

[76] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[77] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[78] M. A. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *IEEE transactions on signal processing*, vol. 57, no. 6, pp. 2275–2284, 2009.

[79] Y. Yu, A. P. Petropulu, and H. V. Poor, "Mimo radar using compressive sampling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 146–163, 2010.

[80] C. R. Berger and J. M. F. Moura, "Noncoherent compressive sensing with application to distributed radar," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, IEEE, 2011, pp. 1–6.

[81] D. Liu, U. S. Kamilov, and P. T. Boufounos, "Sparsity-driven distributed array imaging," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*, IEEE, 2015, pp. 441–444.

[82] H. J. Cho and D. C. Munson, "Overcoming polar-format issues in multichannel sar autofocus," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, IEEE, 2008, pp. 523–527.

[83] I. B. Collings and D. A. Gray, "Deconvolution techniques for non-coherent radar images.," in *ISSPA*, 1996, pp. 113–116.

[84] T Derham, S Doughty, C Baker, and K Woodbridge, "Ambiguity functions for spatially coherent and incoherent multistatic radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, 2010.

[85] W. Ye, T. S. Yeo, and Z. Bao, "Weighted least-squares estimation of phase errors for sar/isar autofocus," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2487–2494, 1999.

[86] K.-H. Liu, A. Wiesel, and D. C. Munson, "Synthetic aperture radar autofocus via semidefinite relaxation," *IEEE transactions on image processing*, vol. 22, no. 6, pp. 2317–2326, 2013.

[87] ——, "Synthetic aperture radar autofocus based on a bilinear model," *IEEE Transactions on image Processing*, vol. 21, no. 5, pp. 2735–2746, 2012.

[88] D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, "Phase gradient autofocus-a robust tool for high resolution sar phase correction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 827–835, 1994.

[89] N. O. Onhon and M. Cetin, "A sparsity-driven approach for joint sar imaging and phase error correction," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2075–2088, 2012.

[90] H. Mansour, D. Liu, U. S. Kamilov, and P. T. Boufounos, "Sparse blind deconvolution for distributed radar autofocus imaging," *arXiv preprint arXiv:1805.03269*, 2018.

[91] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1711–1732, 2014.

[92] Y. Li, K. Lee, and Y. Bresler, "Optimal sample complexity for blind gain and phase calibration.," *IEEE Trans. Signal Processing*, vol. 64, no. 21, pp. 5549–5556, 2016.

[93] X. Du, C. Duan, and W. Hu, "Sparse representation based autofocusing technique for isar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 3, pp. 1826–1835, 2013.

[94] H. Mansour, D. Liu, P. T. Boufounos, and U. S. Kamilov, "Radar autofocus using sparse blind deconvolution," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 1623–1627.

[95] M. A. Lodhi, H. Mansour, and P. T. Boufounos, "Coherent radar imaging using unsynchronized distributed antennas," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 4320–4324.

[96] D. Liu, G. Kang, L. Li, Y. Chen, S. Vasudevan, W. Joines, Q. H. Liu, J. Krolik, and L. Carin, "Electromagnetic time-reversal imaging of a target in a cluttered environment," *IEEE transactions on antennas and propagation*, vol. 53, no. 9, pp. 3058–3066, 2005.

[97] Y. Li, K. Lee, and Y. Bresler, "Identifiability in bilinear inverse problems with applications to subspace or sparsity-constrained blind gain and phase calibration," *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 822–842, 2017.

[98] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[99] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[100] S. Ling and T. Strohmer, "Self-calibration and biconvex compressive sensing," *Inverse Problems*, vol. 31, no. 11, p. 115 002, 2015.

[101] A. Ahmed and L. Demanet, "Leveraging diversity and sparsity in blind deconvolution," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 3975–4000, 2018.

[102] K. Greenewald, S. Zhou, and A. Hero III, "Tensor graphical lasso (teralasso)," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 81, no. 5, pp. 901–931, 2019.

[103] M. A. Lodhi and W. U. Bajwa, "Learning product graphs underlying smooth graph signals," *arXiv preprint arXiv:2002.11277*, 2020.

[104] D. T. Kaplan, "Statistical modeling: A fresh approach," 2009.

[105] M. Rudelson and R. Vershynin, "On sparse reconstruction from fourier and gaussian measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 61, no. 8, pp. 1025–1045, 2008.